RFSPKSSM: SPIKING STATE SPACE MODELS FOR EFFICIENT RADAR TARGET RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks and state-space models offer complementary approaches to efficient sequence processing, combining sparse event-driven computation with scalable long-range modelling without quadratic dependence on sequence length. Existing spiking state-space models focus largely on generic sequence modelling, with limited exploration in radar sensing and signal processing. We propose radar-oriented spiking state-space models (RFspkSSM), built around a second-order oscillatory spiking state-space model (OscSpikeSSM) that combines trainable neuronal parameterisations with stability-constrained oscillatory dynamics. For radar sensing, the core extends to structured point clouds, multi-view heatmaps, and radar-IMU representations spanning applications such as activity and gesture recognition, 3D human and hand pose estimation, and raw FMCW signal processing. Across 10 benchmark datasets, RFspkSSM efficiently outperform baselines. More specifically, RFspkSSM attains 96.8% accuracy in radar gesture recognition, reduces pose centre-distance error to 9.2 cm ($3.41\times$ lower), uses up to $125\times$ fewer parameters for mmWave recognition, and achieves up to $3.6\times$ lower theoretical compute energy in radar signal processing. Our results show that RFspkSSM generalises across heterogeneous sequence and radar representations, combining strong task performance with efficient modelling that learns spectral processing from sequence data.

1 INTRODUCTION

Millimetre-wave radar has become an attractive sensing modality for human perception because it provides motion and geometric information while remaining robust to illumination and preserving greater privacy than camera-based sensing. Its use in gesture recognition, activity understanding, and pose estimation, however, requires models that can capture temporal dependencies while remaining suitable for continuously operating, resource-constrained systems. State-space models (SSMs) provide a natural framework for this setting. SSMs have emerged as scalable alternatives to attention-based sequence models by representing temporal dependencies via recurrent states while avoiding the quadratic dependence on sequence length of standard self-attention (Gu et al., 2022; Smith et al., 2023; Gu & Dao, 2024). Oscillatory SSMs further introduce second-order dynamics that represent temporal information through both state amplitude and phase (Rusch & Rus, 2025), a potentially useful property for radar signals with inherent temporal and spectral structure.

Spiking neural networks (SNNs) provide a complementary approach based on sparse event-driven communication. Recent work has combined spiking neurons with state-space models for long-range sequence processing (Stan & Rhodes, 2024; Shen et al., 2025; Bal & Sengupta, 2025). Second-order resonate-and-fire neurons additionally introduce frequency-selective oscillatory dynamics (Izhikevich, 2001), and recent spiking state-space models have demonstrated their effectiveness for long-sequence modelling (Agrawal et al., 2026). Presently, R&F neurons, though, are expressive but require larger hardware footprints and fewer neurons per unit area compared to the LIF family of neurons. Moreover, these models have primarily been studied as general sequence architectures. Radar sensor-based applications introduce additional requirements arising from their spatial, geometric, and spectral structure. Existing radar SNNs typically rely on first-order neuronal dynamics or operate on conventionally preprocessed range-Doppler and micro-Doppler representations (Liao et al., 2026; Tsang et al., 2021; Safa et al., 2023). This motivates a broader question: can oscillatory

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

spiking state-space dynamics be utilised for numerous radar perception applications such as gesture recognition, activity tracking, pose estimation or signal processing?

We address this question with *RFspkSSM*, a family of spiking state-space models for radar sensing built around *OscSpikeSSM*, a second-order oscillatory spiking state-space model that combines trainable SiLIF/AdLIF-type neuronal dynamics (Fabre et al., 2025; Yin et al., 2021) with LinOSS-inspired oscillatory state evolution (Rusch & Rus, 2025). The OscSpikeSSM core couples stable temporal dynamics with sparse spike-based communication. We adapt this core to heterogeneous radar modalities and to frequency-selective processing of raw FMCW measurements. Conventional FMCW radar pipelines typically construct range-Doppler representations via fixed spectral transforms (Müller et al., 2023); our formulation instead incorporates oscillatory spiking state-space dynamics into the radar processing pipeline. We evaluate the resulting RFspkSSM family across long-range time-series classification and diverse radar perception tasks, including benchmarks from the UEA archive (Bagnall et al., 2018) and mmWave datasets spanning point-cloud, multi-view, radar-IMU, and range-Doppler sensing (Liao et al., 2026; Rahman et al., 2024; Lv et al., 2025; Lien et al., 2016). Our core contributions include: (1) RFspkSSM, a family of radar-oriented spiking state-space models built around OscSpikeSSM, a second-order oscillatory spiking core with stability-constrained underdamped dynamics, subtractive state reset, and spike-coupled feedback; (2) Radar-specific RFspkSSM architectures for mmWave activity and gesture recognition, multi-view 3D human pose estimation, radar-IMU hand pose estimation, and online radar gesture classification; (3) a frequency-selective oscillatory front end for raw FMCW radar, where trainable spiking state-space dynamics replace conventional range and Doppler FFT processing for radar gesture recognition and classification; (4) a comprehensive empirical evaluation of RFspkSSM against spiking, state-space, and radar-specific baselines, together with ablations on the oscillatory dynamics, spike reset, and task-specific model components.

2 Related Works

Radar Sensing and Deep Learning: Learning-based radar sensing has evolved from CNN-based processing toward attention, Transformer, and state-space architectures for structured spatial and temporal modelling. Early works demonstrated RF-based 3D human pose estimation (Zhao et al., 2018) and CNN-based skeletal reconstruction from mmWave radar (Sengupta et al., 2020), while HuPR introduced attention-based multi-scale feature fusion for radar pose estimation (Lee et al., 2023). More recent approaches use transformers for end-to-end radar sensing (Zheng et al., 2024) and cross-view 3D human pose estimation (Kato et al., 2025), while mmEgoHand combines mmWave radar with inertial measurements for hand pose and gesture estimation (Lv et al., 2025). RFMamba further introduces state-space modelling to RF perception through frequency-aware spatio-temporal processing of long RF sequences (Zhang et al., 2025). However, these approaches rely on dense CNN, transformer, or continuous-valued SSM backbones and do not unify spike-based state-space dynamics with structured radar representations and learnable FMCW spectral processing.

State-Space and Oscillatory Sequence Models: Structured state-space models enable scalable long-range sequence modelling through recurrent state dynamics with efficient parallel training. S4 introduced continuous state-space layers (Gu et al., 2022), followed by simplified formulations such as S5 (Smith et al., 2023) and the Linear Recurrent Unit (LRU) (Orvieto et al., 2023). Mamba further introduced input-dependent selective dynamics with an efficient scan implementation (Gu & Dao, 2024). In parallel, oscillatory sequence models such as coRNN and UnICORNN employ second-order recurrent dynamics to preserve information over long temporal horizons (Rusch & Mishra, 2021a;b), while LinOSS incorporates stable oscillatory dynamics directly into a linear state-space formulation (Rusch & Rus, 2025).

Spiking Neural Networks and Spiking State-Space Models: SNNs represent temporal information through discrete spike events, most commonly using integrate-and-fire or leaky integrate-and-fire neuronal dynamics. Resonate-and-fire neurons extend this formulation with subthreshold oscillations and frequency-selective responses (Izhikevich, 2001). Recent work has connected spiking dynamics with state-space modelling through learnable multi-state neuronal transitions (Fabre et al., 2025) and long-range spiking sequence models (Stan & Rhodes, 2024). SpikingSSMs combines spiking dynamics with structured SSMs using surrogate dynamics for parallel training (Shen et al.,

2025), while P-SpikeSSM employs probabilistic spike generation for parallelizable long-range modelling (Bal & Sengupta, 2025). Second-order resonate-and-fire dynamics have also recently been incorporated into spiking SSMs for long time-series modelling and signal-processing tasks (Agrawal et al., 2026). Separately, SNNs have been explored for radar gesture and activity recognition (Tsang et al., 2021; Safa et al., 2023; Liao et al., 2026), but typically rely on first-order neuronal dynamics or precomputed range-Doppler and micro-Doppler representations. In contrast, RFspkSSM extends spiking state-space dynamics across structured radar perception and raw FMCW processing, supporting point-cloud, multi-view, radar-IMU, and streaming inputs while also providing a frequency-selective alternative to conventional range and Doppler FFT processing.

3 METHODS

3.1 OSCILLATORY SPIKING STATE-SPACE MODEL (OSCSPIKESSM)

We develop an oscillatory spiking state-space model, referred to as *OscSpikeSSM*, by combining the trainable neuronal parameterisation of SiLIF/AdLIF-type spiking models (Fabre et al., 2025) with the second-order oscillatory dynamics of LinOSS (Rusch & Rus, 2025). The two models contribute different parts of the formulation. SiLIF/AdLIF provides heterogeneous, trainable temporal rates; spike-dependent adaptation and reset; and threshold-based spike communication between layers. LinOSS provides a second-order latent state in which temporal information is represented through damped oscillatory trajectories. We couple these two components through a constrained state-space parameterisation that keeps every hidden mode in the underdamped regime while retaining a binary spiking interface.

Motivation and Design Rationale: Our main goal is not only to optimise for long generic sequence benchmarks, but also to account for the temporal and spectral structure encountered in radar sensing. First-order spiking dynamics can lose information through exponential membrane decay (Stan & Rhodes, 2024), whereas second-order oscillatory states preserve information through oscillatory evolution, making them suitable for periodic motion, resonant responses, and temporally structured radar measurements (Rusch & Rus, 2025). Direct resonate-and-fire dynamics, however, can exhibit divergent membrane responses, excessive spiking, and reset-induced disruption of oscillations (Higuchi et al., 2024). Such uncontrolled dynamics can increase spike activity and state variability, making hardware implementation more difficult for resource-constrained radar sensing systems (Arsalan et al., 2022). We therefore combine stability-constrained oscillatory dynamics with trainable neuronal parameters, subtractive reset, and spike-dependent feedback to obtain long-lived but bounded state evolution. The resulting OscSpikeSSM combines bounded oscillatory memory with sparse spike-based communication to capture radar-specific temporal and spectral structure while remaining suitable for resource-constrained implementations. This approach also makes it amenable for potential implementation in analog spiking networks(El-Masry et al., 2024; Moradi et al., 2018).

Trainable Neuronal Parameterization: For an input sequence $\mathbf{x}_{1:T}$, with $\mathbf{x}_t \in \mathbb{R}^d$, each timestep is first projected to the hidden dimension as $\mathbf{I}_t = \mathbf{W}_{\text{in}}\mathbf{x}_t + \mathbf{b}_{\text{in}}$, where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{h\times d}$, $\mathbf{b}_{\text{in}} \in \mathbb{R}^h$, and $\mathbf{I}_t \in \mathbb{R}^h$ denotes the input current supplied to the state-space dynamics. This is followed by batch or layer normalisation. Following the SiLIF/AdLIF parameterization, the temporal rates and integration step $\lambda_\alpha, \lambda_\beta, \Delta t \in \mathbb{R}^h$ are learned independently for each hidden unit in the log domain as $\lambda_\alpha = \exp(\lambda_{\alpha,\log})$, $\lambda_\beta = \exp(\lambda_{\beta,\log})$, and $\Delta t = \exp(\Delta t_{\log})$. This parameterisation ensures that the decay rates and timestep remain positive during training. We additionally retain two AdLIF-inspired coupling parameters, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^h$, which control the restoring dynamics and spike-dependent feedback, respectively. In the implementation, they are bounded during the forward pass as $a_i \in [0, 1]$ and $b_i \in [0, 2]$. The resulting layer, therefore, inherits heterogeneous neuronal timescales and spike-dependent regulation from the SiLIF/AdLIF formulation rather than using a single shared decay constant.

LinOSS-inspired second-order state space: Following the oscillator-based state-space formulation of LinOSS (Rusch & Rus, 2025), each hidden unit maintains two continuous variables, $\mathbf{u}(t), \mathbf{v}(t) \in \mathbb{R}^h$. Before introducing the spiking terms, their continuous-time dynamics can be written as $\mathbf{u}'(t) = \mathbf{v}(t),\ \mathbf{v}'(t) = -c\mathbf{v}(t) - k\mathbf{u}(t) + B\mathbf{I}(t)$ where $\mathbf{u}(t), \mathbf{v}(t) \in \mathbb{R}^h$ are the position-like and velocity-like states, respectively, $c, k, B \in \mathbb{R}^h$ denote the per hidden unit damping, restoring, and input-coupling coefficients, and $\mathbf{I}(t) \in \mathbb{R}^h$ denotes the projected input supplied to the state-

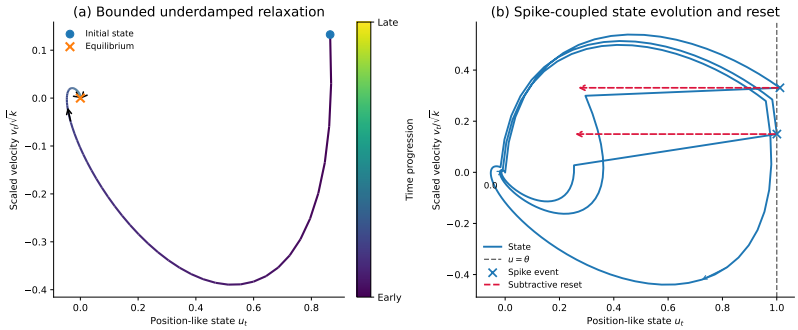

Figure 1: Phase-space dynamics of OscSpikeSSM. The free second-order state exhibits bounded underdamped relaxation toward equilibrium, while spike-coupled dynamics preserve the velocity-like state across subtractive resets and retain the oscillatory trajectory.

space layer. The continuous state-space system is $\begin{bmatrix}\mathbf{u}'(t)\\ \mathbf{v}'(t)\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{K} & -\mathbf{C}\end{bmatrix}\begin{bmatrix}\mathbf{u}(t)\\ \mathbf{v}(t)\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{B}\end{bmatrix}\mathbf{I}(t)$ with $\mathbf{C} = \mathrm{diag}(c) \in \mathbb{R}^{h\times h}$, $\mathbf{K} = \mathrm{diag}(k) \in \mathbb{R}^{h\times h}$, and $\mathbf{B} = \mathrm{diag}(B) \in \mathbb{R}^{h\times h}$, while $\mathbf{I} \in \mathbb{R}^{h\times h}$ denotes the identity matrix. The hidden modes evolve independently after the learned input projection, giving a structured second-order recurrence rather than a dense recurrent hidden-to-hidden transformation. This is the part of the model inherited from the oscillatory state-space formulation of LinOSS.

Coupling the spiking and oscillatory parameterisations: The SiLIF/AdLIF parameters are mapped to the oscillator coefficients $c, B, R \in \mathbb{R}^h$ as $c = \lambda_\alpha + \lambda_\beta$, $B = \lambda_\alpha\lambda_\beta$, and $R = \lambda_\alpha\lambda_\beta b$. The restoring coefficient $k \in \mathbb{R}^h$ is constrained through the oscillation-safe parameterization as $k_{\mathrm{raw}} = \lambda_\alpha\lambda_\beta(1+a)$ and $k = \max\left(k_{\mathrm{raw}}, \frac{c^2}{4} + \epsilon\right)$. Here, the maximum is applied element-wise and $\epsilon > 0$ prevents the dynamics from reaching the critically damped boundary. For an individual hidden mode, the homogeneous continuous-time dynamics are obtained by setting the input term to zero, giving $u'(t) = v(t)$ and $v'(t) = -cv(t) - ku(t)$. Eliminating $v(t)$ yields the second-order differential equation $u''(t) + cu'(t) + ku(t) = 0$. Assuming an exponential solution of the form $u(t) = e^{rt}$, gives $r^2 + cr + k = 0$ with eigenvalues $r = -\frac{c}{2} \pm j\sqrt{k - \frac{c^2}{4}}$. Since $c > 0$ and the parameterisation enforces $k > c^2/4$, each mode has a negative real part and a non-zero imaginary component. The imposed lower bound on k therefore keeps the underlying continuous-time dynamics in a stable underdamped regime.

Oscillatory state dynamics and Discretisation: Building on the damped second-order oscillator above, we incorporate spike-dependent feedback into the state evolution. For each hidden mode, the resulting continuous-time dynamics are $u'(t) = v(t),\ v'(t) = -c\,v(t) - k\,u(t) + B\,I(t) - R\,s(t)$. The additional term $-R\,s(t)$ feeds the emitted spike $s(t) \in \{0,1\}$ back into the oscillator dynamics, so that a firing event influences the subsequent evolution of the latent state. We discretise the spike-coupled oscillator using a damped IMEX scheme (Rusch & Rus, 2025), with implicit damping and explicit restoring, input, and spike-dependent terms. Subtractive reset acts directly on the position-like state; full update equations are given in Appendix A.1.2.

Spike generation and layer interface: After the state update, the position-like state is converted into a binary spike vector $\mathbf{s}_t \in \{0,1\}^h$ according to $\mathbf{s}_t = H(\mathbf{u}_t - \theta)$ where $H(\cdot)$ denotes the Heaviside step function and θ is the firing threshold. Since the threshold operation is non-differentiable, surrogate gradients are used during backpropagation while retaining hard binary spikes in the forward pass (Neftci et al., 2019). The implementation uses boxcar and multi-Gaussian surrogate functions (Bittar & Garner, 2022; Yin et al., 2021) to provide gradients around the firing threshold. The choice of surrogate affects only gradient propagation; the forward spike generation remains governed by the hard threshold. For stacked OscSpikeSSM layers, the spike sequence produced by one layer forms the input to the next. In the bidirectional setting, the layer processes the sequence in both temporal directions; the reverse output is restored to its original order and concatenated with the forward spike representation, yielding an output dimension of $2h$ rather than h.

Phase-Space Dynamics: Figure 1 illustrates two key properties of OscSpikeSSM. The free-state trajectory remains bounded and decays toward equilibrium, showing how the second-order dynamics preserve oscillatory structure while remaining stable. When spikes are introduced, the reset acts only on the position-like state, while the velocity-like state is retained and continues to shape the subsequent trajectory. This coupling between oscillatory state evolution and spike-dependent regulation distinguishes OscSpikeSSM from first-order membrane dynamics and allows temporal information to persist across firing events.

4 RESULTS

4.1 LONG RANGE CLASSIFICATION

We begin by evaluating the proposed core spiking SSM for long-sequence classification using EigenWorms, EthanolConcentration, and MotorImagery datasets. The input sequence is processed directly by the OscSpikeSSM core and the classification readout. The full architectural specifics and implementation details are provided in Appendix A.2.2. Although the main focus of this work is on developing FMCW radar-specific spiking SSM architecture, from Table 9 we notice that our design (OscSpkSSM) outperforms (57.8%) the other spiking SSM designs, P-SpikeSSM and SpikingSSM (Bal & Sengupta, 2025; Shen et al., 2025)(56.3% and 54.3% respectively). Above all, OscSpkSSM achieves this without heavy nonlinearities such as GELU/GLU, unlike P-SpikeSSM and SpikingSSM.

4.2 MMWAVE SENSING WITH OSCSPKSSM

To examine the behaviour of the point-cloud variant of the OscSpkSSM family across different mmWave recognition tasks, we evaluate *mm-OscSpikeSSM* on the publicly available AOPHand, mmFiT, and MMActivity datasets (Zafar, 2023; Tiwari & Gupta, 2021; Singh et al., 2019). The three datasets cover hand-gesture, fitness-activity, and human-activity recognition, respectively, and differ considerably in class structure and sample availability. mm-OscSpikeSSM uses a common mmWave point-cloud representation and the shared OscSpikeSSM temporal core, while adapting only the spatial and temporal configurations to each task. We further evaluate the proposed OscSpikeSSM core on the Google Soli dataset (Lien et al., 2016) to study its behaviour on short radar sequences with fine-grained dynamic gestures. These evaluations, therefore, test whether the same OscSpikeSSM design can generalise across heterogeneous mmWave sensing settings without introducing a task-specific sensing backbone. Please refer to Appendix A.2.3 for a detailed explanation of architecture and evaluation protocols.

We compare mm-OscSpikeSSM against the mmWave recognition baselines reported by Liao et al. (2026), which span feed-forward, convolutional, recurrent, hybrid, and spiking models under the same four-frame, 64-point representation. Using top-1 accuracy as the primary metric, Table 1 shows that mm-OscSpikeSSM achieves the best results on mmFiT and MMActivity, while remaining within 0.60 percentage points of the best AOPHand result. It also attains the highest average accuracy (86.72%) with only 0.14–0.26M parameters, corresponding to roughly 68–$126\times$ fewer parameters than SpikingResFormer. These results are consistent with the lightweight frame-level spatial encoder and OscSpikeSSM dynamics providing temporal integration without a large spiking convolutional backbone. Table 2 shows that, compared with SpikingResFormer, mm-OscSpikeSSM reduces model memory by approximately 68–$96\times$ and full-test inference time by 12–$14\times$. On mmFiT, theoretical energy is reduced by $1.41\times$, whereas AOPHand incurs a $1.50\times$ energy tradeoff despite substantially lower memory usage and inference latency.

For Google Soli, each gesture is represented as a 40-frame range-Doppler sequence and processed using two stacked OscSpikeSSM layers followed by a leaky temporal readout. Unlike the reference SRNN, which relies on recurrent LIF/ALIF dynamics (Yin et al., 2021), temporal memory is carried by the OscSpikeSSM dynamics while sparse spikes provide the inter-layer representation. The full architectural specifics and implementation details are provided in Appendix A.2.4. Table 3 shows that Soli-OscSpikeSSM improves sequence accuracy over SRNN by 4.9 percentage points while using $3.0\times$ less model memory, $1.61\times$ lower inference time, and $2.0\times$ lower theoretical energy. These gains are consistent with the compact OscSpikeSSM temporal backbone, which conveys sequence information via oscillatory state dynamics while retaining sparse, spike-based communication.

Table 1: Performance on AOPHand, mmFiT, and MMActivity datasets (refer Table 10 for extended version). Reference results are reported as mean ± standard deviation over three runs, while our results are reported over five runs with different seeds. Text in bold and underlined denotes the first- and second-best accuracies and the smallest and second-smallest SNN models.

	AOPHand		mmFiT		MMActivity		
Baseline	**Accuracy (%)**	**Params (M)**	**Accuracy (%)**	**Params (M)**	**Accuracy (%)**	**Params (M)**	**Avg**
VGG9	$74.39_{\pm 0.61}$	31.603	$69.36_{\pm 2.15}$	31.605	$70.00_{\pm 4.33}$	31.601	71.25
ResNet18	$72.47_{\pm 1.30}$	11.173	$71.94_{\pm 1.29}$	11.174	$61.67_{\pm 6.29}$	11.173	68.69
ResNet101	$69.70_{\pm 0.72}$	42.506	$72.64_{\pm 0.25}$	42.508	$64.17_{\pm 1.44}$	42.504	68.84
MS-ResNet50	$89.51_{\pm 0.69}$	23.514	$85.41_{\pm 0.83}$	23.516	$67.50_{\pm 4.33}$	23.512	80.81
SpikFormer	$91.29_{\pm 1.58}$	2.566	$83.58_{\pm 0.52}$	2.567	$\underline{75.83}_{\pm 1.44}$	2.566	$\underline{83.57}$
SpikingResFormer	$\mathbf{92.74}_{\pm \mathbf{0.64}}$	17.310	$\underline{86.38}_{\pm 0.74}$	17.310	$65.83_{\pm 3.82}$	17.309	81.65
mm-OscSpikeSSM	$\underline{92.14}_{\pm 0.32}$	$\mathbf{0.182}_{\downarrow 95.11\times}$	$\mathbf{90.02}_{\pm \mathbf{0.52}}$	$\mathbf{0.255}_{\downarrow 67.88\times}$	$\mathbf{78.00}_{\pm \mathbf{1.87}}$	$\mathbf{0.138}_{\downarrow 125.42\times}$	**86.72**

Table 2: Efficiency comparison on AOPHand and mmFiT using the metrics and inference protocols defined in Appendix A.2.1. Subscripts report the relative change with respect to SpikingResFormer, with ↓/↑ denoting lower/higher values.

	AOPHand			mmFiT		
Model	**Memory** (MiB)	**Energy** (μJ/sample)	**Wall Clock** (s)	**Memory** (MiB)	**Energy** (μJ/sample)	**Wall Clock** (s)
SpikingResFormer	66.23	435.84	7.95 ± 0.15	66.23	372.92	9.69 ± 0.09
mm-OscSpikeSSM	$0.69_{\downarrow 95.98\times}$	$654.93_{\uparrow 1.50\times}$	$0.67 \pm 0.02_{\downarrow 11.86\times}$	$0.97_{\downarrow 68.27\times}$	$263.63_{\downarrow 1.41\times}$	$0.69 \pm 0.03_{\downarrow 14.04\times}$

4.3 Radar based 3D Pose Estimation using Spiking SSM

Having established the OscSpikeSSM core on generic sequences and smaller radar recognition tasks, we next develop application-specific architectures for more structured radar perception problems. For 3D pose estimation, we use RAPTR (Kato et al., 2025) as a recent representative baseline that processes both horizontal and vertical radar heatmaps via Transformer-based cross-view fusion and a two-stage pose decoder. Our model addresses the same setting with OscSpikeSSM-based view-specific processing followed by depth-aware cross-view fusion, allowing us to examine whether oscillatory spiking state-space representations can replace attention-based multi-view modelling. We evaluate this formulation on the MMVR dataset, which represents each sample using paired horizontal-depth and vertical-depth mmWave radar heatmaps for 3D human pose estimation. The complementary views provide distinct spatial information while sharing a common depth axis.

MMVR-OscSpikeSSM Architecture (Figure 2): Given horizontal-depth and vertical-depth radar heatmaps, $\mathbf{X}_{\text{hor}} \in \mathbb{R}^{B\times T\times W\times D}$ and $\mathbf{X}_{\text{ver}} \in \mathbb{R}^{B\times T\times H\times D}$, the two views are independently embedded and processed by a two-stage encoder composed of RF-SpikingOSS blocks with depth-aware cross-view fusion. Each block combines a heatmap and a spatio-temporal branches built around the shared OscSpikeSSM core. To preserve the radar geometry, the state-space dynamics scan bidirectionally along the temporal, spatial, and depth axes, $\mathcal{D} = \{T^+, T^-, S^+, S^-, D^+, D^-\}$, where S denotes the horizontal or vertical spatial dimension. The two views are then combined through their shared depth axis, while the second encoder stage operates at reduced spatial and depth resolution before pose decoding. The fused representation is passed to a query-based decoder that generates and refines candidate 3D poses with confidence estimates. Training is weakly supervised, without fine-grained 3D keypoint annotations; matched queries are supervised using 3D geometric constraints, projected 2D keypoint and Object Keypoint Similarity (OKS) losses, bone-structure regularisation, and confidence supervision. We evaluate on the official MMVR P1S1 split using four synchronised horizontal-depth and vertical-depth radar frames per sample. Performance is reported using 3D box-center distance and horizontal, vertical, and depth edge errors in centimetres. Please refer to Appendix A.2.5 for a detailed description of MMVR-OscSpikeSSM.

Tables 4 and 5 show that this MMVR-OscSpikeSSM substantially improves 3D pose estimation over RAPTR, reducing center-distance error by $3.41\times$ and the vertical and depth edge errors by $2.62\times$ and $2.82\times$, respectively, while maintaining comparable horizontal-edge error. It also improves center, vertical, and depth errors over MMVR-P-SpikeSSM, with a small tradeoff in the

Table 3: Gesture-recognition performance on the Google Soli dataset. For Soli-OscSpikeSSM, accuracy is reported as mean ± standard deviation over five training runs with different seeds. Computational efficiency metrics and inference protocols are defined in Appendix A.2.1. Subscripts report the relative change with respect to SRNN, with ↓/↑ denoting lower/higher values.

Model	Sequence Acc. (%)	Model Memory (MiB)	Inference (ms/sample)	Energy (mJ/sample)
SRNN	91.9	4.0	13.7	97.5
Soli-OscSpikeSSM	96.8 ± 0.3	$1.3_{\downarrow 3.0\times}$	$8.5_{\uparrow 1.6\times}$	$48.4_{\downarrow 2.0\times}$

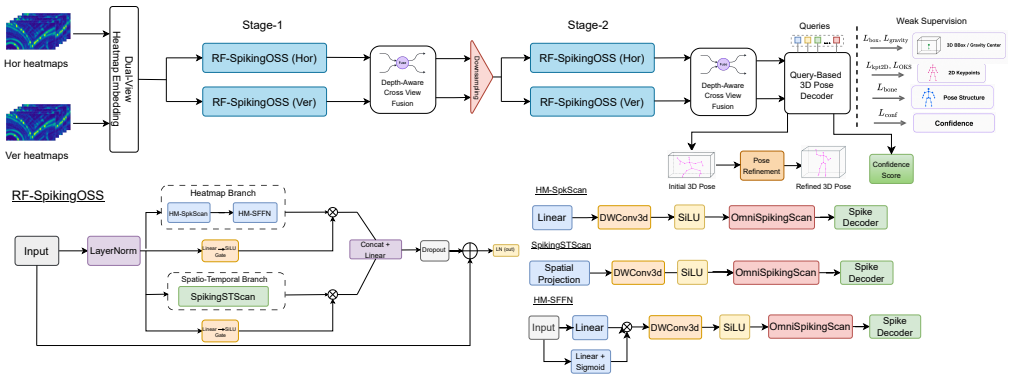

Figure 2: Overview of the proposed MMVR-OscSpikeSSM architecture for weakly supervised 3D human pose estimation.

horizontal dimension. MMVR-P-SpikeSSM retains the probabilistic spiking SSM backbone of P-SpikeSSM, with certain modifications to the input and output interfaces adapted to MMVR, treating the dual-view heatmaps as a generic token sequence. In contrast, MMVR-OscSpikeSSM explicitly preserves the two radar views through axis-aware spiking scans, depth-aware cross-view fusion, and query-based pose refinement, which is consistent with the improved pose estimates. Compared with RAPTR, MMVR-OscSpikeSSM uses $6.37\times$ less model memory and $1.61\times$ lower theoretical compute energy, consistent with its more compact architecture and sparse spike-driven computation. This efficiency comes with a modest $1.29\times$ increase in wall clock time on the graphics processor. The ablation study further isolates the contribution of the recurrent core and scan configuration. Replacing OscSpikeSSM with a conventional LIF core increases the center and depth errors from 9.20 and 17.92, cm to 11.35 and 18.64, cm, respectively, providing the clearest evidence for the contribution of the proposed second-order oscillatory dynamics. The directional ablations show a more axis-dependent behaviour: forward-only scanning yields $8.48/22.41/4.00/17.11$, cm for center/h/v/d error, while temporal-only scanning gives $8.43/20.87/4.64/17.11$, cm. These results indicate that the oscillatory core provides the clearest contribution, whereas bidirectional and multi-axis scanning produce a more balanced geometric representation across axes rather than uniformly minimising every individual error.

4.4 Performance on Ego-Centric Hand Pose Estimation

We next extend our spiking SSM backbone to dense radar-IMU hand-pose estimation (Lv et al., 2025), where accurate localisation requires preserving fine spatial evidence within each radar frame together with temporal consistency across consecutive observations. The model combines a dense radar encoder, spatially resolved spiking SSM processing, IMU-guided radar conditioning, query-based hand discovery and refinement, and a structured 21-joint prediction head. Rather than reducing each radar frame into a single vector, the model retains spatial radar tokens and applies a shared bidirectional spiking SSM along the temporal trajectory of each location. This preserves localised radar evidence while introducing temporal modelling at each spatial position. A wrist-prior branch adds coarse localisation cues, while the IMU stream is encoded separately and used to condition the radar features with motion information. Multi-hand prediction is handled with learnable hand queries that are refined over time and decoded using a structured hand model. The decoder pre-

Table 4: Pose estimation performance (mean and standard deviation over three training runs) on MMVR. We use the P1S1 data split, which represents a single-person scenario in an open-space environment (Kato et al., 2025; Wan et al., 2024; Ho et al., 2024; Yan et al., 2024). Subscripts report the relative change with respect to RAPTR with ↓ denoting lower values.

Method	Center distance (cm)	Edge length error (cm)		
		(h)	(v)	(d)
Person-in-WiFi 3D	136.14	33.18	95.43	242.86
QRFPose	210.75	38.12	73.69	409.38
HRRadarPose	164.46	37.84	74.00	313.81
RAPTR	31.41	22.90	10.66	50.56
MMVR-P-SpikeSSM	18.36 ± 0.24	20.92 ± 0.03	4.69 ± 0.07	20.20 ± 0.26
MMVR-OscSpikeSSM (ours)	$9.20 \pm 0.59_{\downarrow 3.41\times}$	$22.26 \pm 0.41_{\downarrow 1.03\times}$	$4.06 \pm 0.17_{\downarrow 2.62\times}$	$17.92 \pm 0.39_{\downarrow 2.82\times}$

Table 5: Computational efficiency on the MMVR P1S1 setting, using the metrics and inference protocols defined in Appendix A.2.1. Subscripts show relative changes from RAPTR, with ↓/↑ denoting lower/higher values.

Model	Model Memory (MiB)	Inference (ms/sample)	Energy (mJ/sample)
RAPTR	27.87	154.86	35.90
MMVR-OscSpikeSSM	$4.37_{\downarrow 6.37\times}$	$200.21_{\uparrow 1.29\times}$	$22.29_{\downarrow 1.61\times}$

dicts the wrist together with bone directions and lengths, followed by bounded joint residuals, rather than regressing all joint coordinates independently. Query assignments are maintained consistently across the sequence, allowing the model to preserve hand identity during temporal pose estimation. These components extend the spiking SSM model beyond a single temporal backbone by using the same state-space formulation for spatially resolved radar dynamics, IMU motion encoding, and hand-instance refinement.

We evaluate the mode on the EgoHands dataset, which provides synchronised mmWave radar and IMU measurements, as well as 3D hand keypoint annotations. Please refer to Appendix A.2.6 for a detailed explanation of architecture and evaluation protocols. Table 6 shows that EgoHands-OscSpikeSSM achieves the lowest MPJPE, improving over the strongest mmEgoHand (mmWave+IMU) baseline by 0.87 mm and over the mmWave-only variant by 24.54 mm. This improvement is consistent with the preservation of spatial radar tokens, IMU-guided conditioning, and structured query-based hand decoding. At the same time, EgoHands-OscSpikeSSM uses $21.53\times$ fewer parameters and model memory, and requires $1.53\times$ lower theoretical compute energy. The energy reduction is primarily enabled by the Spiking SSM backbone, which reduces dense MAC processing and shifts part of the sequence computation to sparse, spike-driven AC operations.

4.5 Radar Signal Processing

We next develop and evaluate RF-SpecSpkSSM, a radar signal-processing architecture that replaces conventional range and Doppler FFT stages with oscillatory spiking state-space dynamics. Using the Aircraft Marshalling dataset (Müller et al., 2023), whose raw FMCW measurements are publicly available, we study both the signal-processing front end and the complete recognition pipeline. The goal is to test whether these fixed spectral transforms can be replaced while preserving the range-Doppler structure required for downstream gesture classification. Conventional FMCW processing obtains range-Doppler representations through Fourier transforms along the fast- and slow-time dimensions; we replace these operations with a learnable oscillatory spiking state-space front end trained directly from task labels. The remaining signal conditioning is kept close to the conventional radar pipeline to isolate the effect of the learned spectral processing. We then build an end-to-end spiking, state-space-based architecture that processes the learned range-Doppler sequence and models its temporal evolution for marshalling gesture classification. Please refer to Appendix A.2.7 for implementation details.

From Table 7 we observe that RF-SpecSpkSSM achieves a compact radar-processing pipeline with only 0.6M parameters and 2.4 MiB of model memory. Compared with EfficientNet-B1 (Vermunt & Corradi, 2024), this corresponds to approximately $10.8\times$ lower parameter storage and $3.6\times$ lower theoretical compute energy, while incurring a modest $1.4\times$ increase in inference latency. Despite

Table 6: Comparison of EgoHands-OscSpikeSSM with existing methods on EgoHands (Lv et al., 2025). (a) MPJPE for 3D hand-pose estimation, averaged over two runs. (b) Computational efficiency is compared with mmEgoHand using the metrics and inference protocols defined in Appendix A.2.1. Subscripts show relative changes from mmEgoHand, with ↓ denoting lower values.

(a) 3D Hand Pose Estimation		(b) Model Efficiency		
Method	**MPJPE↓**	**Metric**	**mmEgoHand**	**EgoHands-OscSpikeSSM**
mm4Arm (Liu et al., 2022) (mmWave+IMU)	165.19 mm	**Parameters (M)**	293.11	$13.61_{\downarrow 21.53\times}$
mmEgoHand (w/o Context Decoder)	109.02 mm	**Memory (MiB)**	1118.13	$51.92_{\downarrow 21.53\times}$
mmEgoHand (mmWave)	96.42 mm	**Energy (mJ/clip)**	78.20	$51.06_{\downarrow 1.53\times}$
mmEgoHand (mmWave+IMU)	72.73 mm	**Latency (ms/clip)**	40.95 ± 0.19	165.73 ± 1.44
EgoHands-OscSpikeSSM (Ours)	**71.86 mm**			

Table 7: Classification performance (mean ± standard deviation over five seeds) on the Aircraft Marshalling radar dataset (Vermunt & Corradi, 2024). Computational efficiency metrics and inference protocols are defined in Appendix A.2.1. Subscripts show relative changes from EfficientNet-B1 (Vermunt & Corradi, 2024), with ↓/↑ denoting lower/higher values.

Method	**Params. (M)**	**Acc. (%)**	**Memory (MiB)**	**Latency (ms)**	**Energy (mJ)**
ResNet18	11.2	59.1	42.6	1.8	8.5
EfficientNet-B1	6.5	64.6	24.9	3.8	3.3
LeViT-128s	7.0	59.5	26.8	2.9	0.4
LeViT-128	8.4	58.6	32.2	3.9	2.0
MogaNet Tiny	4.9	66.9	18.9	7.5	5.3
EfficientNet-B1 (Vermunt & Corradi, 2024)	6.5	71.4	24.7	3.7	3.3
FFT-SpikingSSM (Ours)	**0.6**	**64.5 ± 1.1**	**2.3**	**4.9**	**1.1**
RF-SpecSpkSSM (Ours)	$\mathbf{0.6}_{\downarrow 10.8\times}$	**61.4 ± 2.1**	$\mathbf{2.4}_{\downarrow 10.3\times}$	$\mathbf{5.4}_{\uparrow 1.4\times}$	$\mathbf{0.9}_{\downarrow 3.6\times}$

its substantially smaller model capacity, RF-SpecSpkSSM reaches $61.4 \pm 2.1\%$ accuracy, exceeding ResNet18 and both LeViT variants. FFT-SpikingSSM uses conventional FFT preprocessing, followed by the same OscSpikeSSM classifier, thereby providing a direct reference for evaluating the proposed front end. Replacing the fixed FFT stages with the spiking SSM front end of RF-SpecSpkSSM produces approximate range-Doppler representations with only a 3.1 percentage-point accuracy tradeoff, while lowering the theoretical energy from 1.1 to 0.9 mJ per sample with nearly unchanged model size. This suggests that the oscillatory spiking front end can serve as a compact learned approximation to conventional FFT-based preprocessing, retaining competitive recognition performance. Moreover, this experiment extends the role of the proposed second-order oscillatory dynamics beyond temporal feature modelling: the same state-space principle is first used to form frequency-selective range-Doppler representations from raw radar measurements and subsequently to perform sequence modelling. This unifies learnable radar spectral processing and spiking state-space sequence modelling within a single architecture.

5 Discussion

We propose RFspkSSM, a family of radar-oriented spiking state-space models built around OscSpikeSSM, a second-order spiking SSM with stability-constrained underdamped dynamics, spike-dependent feedback, and subtractive reset. Across long-range time-series classification and heterogeneous radar sensing, the same oscillatory dynamics support point-cloud recognition, gesture classification, multi-view 3D pose estimation, and radar-IMU hand-pose estimation while preserving task-specific spatial and geometric structure. The models achieve strong performance with compact, energy-efficient capacity, including competitive long-sequence modelling, high mmWave recognition accuracy, and effective 3D human and hand-pose estimation in weakly supervised and multi-modal settings. Efficiency results show favourable theoretical compute costs, while inference latency remains architecture and hardware-dependent. Beyond perception, RF-SpecSpkSSM replaces range and Doppler FFTs with compact spectral banks for raw FMCW processing, substantially reducing model size and theoretical energy while retaining competitive classification accuracy. Future work will focus on neuromorphic and edge hardware deployment on Intel Loihi 2 and Pulsar for real-time radar sensing.

AI USE STATEMENT

AI tools were used to assist with code implementation and debugging, and language editing of the manuscript. The authors reviewed all AI-assisted text and take full responsibility for the final content of the manuscript.

ETHICS STATEMENT

This work uses publicly available datasets and does not involve the collection of new human-subject data. We use the public datasets for their intended research purposes while acknowledging possible dataset biases and generalisation limitations. While radar-based human sensing can support useful applications, such technologies may also be misused for intrusive monitoring or tracking. We therefore emphasise responsible use of the proposed methods. Our work explores spiking models with the potential for efficient event-driven computation, thereby contributing to Green AI development, while practical benefits depend on the specific deployment platform.

REPRODUCIBILITY STATEMENT

We support reproducibility by providing implementation details in the Appendix A.2.3, A.2.4, A.2.5, A.2.6, A.2.7. Additional experimental details and source code will be made publicly available upon acceptance.

REFERENCES

Kartikay Agrawal, Vaishnavi Nagabhushana, Abhijeet Vikram, Vedant Sharma, and Ayon Borthakur. A spiking heterogeneous harmonic resonate-and-fire state space model for time series. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=dDGxzaRkxO.

Muhammad Arsalan, Avik Santra, and Vadim Issakov. RadarSNN: A resource efficient gesture sensing system based on mm-Wave radar. *IEEE Transactions on Microwave Theory and Techniques*, 70(4):2451–2461, 2022. doi: 10.1109/TMTT.2022.3148403.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron George Bostrom, Paul Southam, and Eamonn J. Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

M. Bal and A. Sengupta. P-SpikeSSM: Harnessing probabilistic spiking state space models for long-range dependency tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Sf4ep9Udjf.

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, volume 31, pp. 795–805, 2018.

Alexandre Bittar and Philip N. Garner. A surrogate gradient spiking baseline for speech command recognition. *Frontiers in Neuroscience*, 16, 2022. doi: 10.3389/fnins.2022.865897.

Federico Danieli, Pau Rodriguez, Miguel Sarabia, Xavier Suau, and Luca Zappella. ParaRNN: Unlocking parallel training of nonlinear RNNs for large language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Moamen El-Masry, Thilo Werner, Amir Zjajo, and Robert Weigel. Toward efficient system-on-module for design-space exploration of analog spiking neural networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(8):3538–3549, 2024. doi: 10.1109/TCSI.2024.3406522.

Maxime Fabre, Lyubov Dudchenko, Younes Bouhadjar, and Emre Neftci. SiLIF: Structured state space model dynamics and parametrization for spiking neural networks, 2025.

Xavier Gonzalez, Andrew Warrington, Jimmy T. H. Smith, and Scott W. Linderman. Towards scalable and stable parallelization of nonlinear RNNs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Saya Higuchi, Sebastian Kairat, Sander Bohte, and Sebastian Otte. Balanced resonate-and-fire neurons. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=dkdilv4XD4.

W. Daniel Hillis and Guy L. Steele Jr. Data parallel algorithms. *Communications of the ACM*, 29 (12):1170–1183, 1986. doi: 10.1145/7902.7903.

Yuan-Hao Ho, Jen-Hao Cheng, Sheng Yao Kuan, Zhongyu Jiang, Wenhao Chai, Hsiang-Wei Huang, Chih-Lung Lin, and Jenq-Neng Hwang. RT-Pose: A 4d radar tensor-based 3d human pose estimation and localization benchmark. In *European Conference on Computer Vision (ECCV)*, pp. 107–125, 2024. doi: 10.1007/978-3-031-73036-8_7.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14. IEEE, 2014. doi: 10.1109/ISSCC.2014.6757323.

Eugene M. Izhikevich. Resonate-and-fire neurons. *Neural Networks*, 14:883–894, 2001. doi: 10.1016/S0893-6080(01)00078-8.

Sorachi Kato, Ryoma Yataka, Pu Wang, Pedro Miraldo, Takuya Fujihashi, and Petros Boufounos. RAPTR: Radar-based 3d pose estimation using transformer. In *Advances in Neural Information Processing Systems*, 2025.

Trung-Khanh Le, Trong-Tu Bui, and Duc-Hung Le. Modeling and designing of an all-digital resonate-and-fire neuron circuit. *IEEE Access*, 11:62318–62336, 2023. doi: 10.1109/ACCESS.2023.3287994.

Shih-Po Lee, Niraj Prakash Kini, Wen-Hsiao Peng, Ching-Wen Ma, and Jenq-Neng Hwang. HuPR: A benchmark for human pose estimation using millimeter wave radar. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5715–5724, 2023. doi: 10.1109/WACV56688.2023.00567.

Zhenyu Liao, Di Yu, Changze Lv, Wentao Tong, Linshan Jiang, Sijie Ji, Xin Du, Hailiang Zhao, Xiaoqing Zheng, and Shuiguang Deng. Frequency matching in spiking neural networks for mmwave sensing. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=kOgsGq2mfC.

Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics*, 35(4):142:1–142:19, 2016. doi: 10.1145/2897824.2925953.

Yilin Liu, Shijia Zhang, Mahanth Gowda, and Srihari Nelakuditi. Leveraging the properties of mmWave signals for 3d finger motion tracking for interactive IoT applications. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3):1–28, 2022. doi: 10.1145/3570613.

Yizhe Lv, Tingting Zhang, Zhijian Wang, Yunpeng Song, Han Ding, Jinsong Han, and Fei Wang. mmEgoHand: Egocentric hand pose estimation and gesture recognition with head-mounted millimeter-wave radar and IMU. *arXiv preprint arXiv:2501.13805*, 2025.

Saber Moradi, Ning Qiao, Fabio Stefanini, and Giacomo Indiveri. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE Transactions on Biomedical Circuits and Systems*, 12(1):106–122, 2018. doi: 10.1109/TBCAS.2017.2759700.

Leon Müller, Manolis Sifalakis, Sherif Eissa, Amirreza Yousefzadeh, Paul Detterer, Sander Stuijk, and Federico Corradi. Aircraft marshaling signals dataset of FMCW radar and event-based camera for sensor fusion. In *2023 IEEE Radar Conference (RadarConf23)*, pp. 1–6, 2023. doi: 10.1109/RadarConf2351548.2023.10149465.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. doi: 10.1109/MSP.2019.2931595.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 26670–26698. PMLR, 2023.

M. Mahbubur Rahman, Ryoma Yataka, Sorachi Kato, Pu Wang, Peizhao Li, Adriano Cardace, and Petros Boufounos. MMVR: Millimeter-wave multi-view radar dataset and benchmark for indoor perception. In *Computer Vision – ECCV 2024*, pp. 306–322. Springer, 2024.

T. Konstantin Rusch and Siddhartha Mishra. Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations*, 2021a. URL `https://openreview.net/forum?id=F3s69XzWOia`.

T. Konstantin Rusch and Siddhartha Mishra. UnICORNN: A recurrent model for learning very long time dependencies. In *International Conference on Machine Learning*, 2021b.

T. Konstantin Rusch and Daniela Rus. Oscillatory state-space models. In *International Conference on Learning Representations*, 2025.

Ali Safa, Federico Corradi, Lars Keuninckx, Ilja Ocket, André Bourdoux, Francky Catthoor, and Georges G. E. Gielen. Improving the accuracy of spiking neural networks for radar gesture recognition through preprocessing. *IEEE Transactions on Neural Networks and Learning Systems*, 34 (6):2869–2881, 2023. doi: 10.1109/TNNLS.2021.3109958.

Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. mm-Pose: Real-time human skeletal posture estimation using mmWave radars and CNNs. *IEEE Sensors Journal*, 20(17):10032–10044, 2020. doi: 10.1109/JSEN.2020.2991741.

Shuaijie Shen, Chao Wang, Renzhuo Huang, Yan Zhong, Qinghai Guo, Zhichao Lu, Jianguo Zhang, and Luziwei Leng. SpikingSSMs: Learning long sequences with sparse and parallel spiking state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20380–20388, 2025. doi: 10.1609/aaai.v39i19.34245.

Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani B. Srivastava. RadHAR: Human Activity Recognition from Point Clouds Generated through a Millimeter-Wave Radar. In *Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*, pp. 51–56, 2019. doi: 10.1145/3349624.3356768.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023.

Matei-Ioan Stan and Oliver Rhodes. Learning long sequences in spiking neural networks. *Scientific Reports*, 14(1):21957, 2024.

Girish Tiwari and Shalabh Gupta. An mmWave radar based real-time contactless fitness tracker using deep CNNs. *IEEE Sensors Journal*, 21(15):17262–17270, 2021. doi: 10.1109/JSEN.2021.3077511.

Ing Jyh Tsang, Federico Corradi, Manolis Sifalakis, Werner Van Leekwijck, and Steven Latré. Radar-based hand gesture recognition using spiking neural networks. *Electronics*, 10(12):1405, 2021. doi: 10.3390/electronics10121405.

Jim Vermunt and Federico Corradi. FMCW radar signal processing pipeline for aircraft marshalling signals classification. In *2024 21st European Radar Conference (EuRAD)*, pp. 457–460, 2024. doi: 10.23919/EuRAD61604.2024.10734929.

Hong Wan, Ruiyuan Song, Chunyang Xie, Zhi Lu, Qi Chen, Zhi Wu, Dongheng Zhang, Yang Hu, and Yan Chen. QRFPose: Query-based 3d pose estimation using radio signals. In *2024 16th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1271–1277, 2024. doi: 10.1109/WCSP62071.2024.10827777.

Kangwei Yan, Fei Wang, Bo Qian, Han Ding, Jinsong Han, and Xing Wei. Person-in-WiFi 3d: End-to-end multi-person 3d pose estimation with Wi-Fi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 969–978, 2024.

Bojian Yin, Federico Corradi, and Sander M. Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10): 905–913, 2021. doi: 10.1038/s42256-021-00397-w.

Ahtsham Zafar. TI-AWR-6843-AOP Gesture Language Data. Kaggle dataset, 2023.

Rui Zhang, Ruixu Geng, Yadong Li, Ruiyuan Song, Hanqin Gong, Dongheng Zhang, Yang Hu, and Yan Chen. RFMamba: Frequency-aware state space model for RF-based human-centric perception. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=lG9fjBLb6d`.

Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. RF-based 3d skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pp. 267–281, 2018. doi: 10.1145/3230543.3230579.

Zhijie Zheng, Diankun Zhang, Xiao Liang, Xiaojun Liu, and Guangyou Fang. RadarFormer: End-to-end human perception with through-wall radar and transformers. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):18285–18299, 2024. doi: 10.1109/TNNLS.2023.3314031.

A Appendix

This appendix provides a detailed description of the proposed architectures, implementation settings, evaluation procedures, and additional experimental results. Section A.1 presents extended details of the OscSpikeSSM formulation, including its discretisation, neuronal dynamics, and parallel-scan implementation. The subsequent sections describe the task-specific architectures, training configurations, and additional results for long-range sequence classification and mmWave sensing, followed by detailed formulations and evaluation protocols for MMVR-based 3D pose estimation and EgoHands hand-pose estimation. Appendix further describes the Soli-OscSpikeSSM architecture and evaluation setup, including frame-level and early-recognition analyses. Finally, we provide architecture diagrams and additional implementation details for the radar signal-processing formulation.

A.1 Extended Methods

A.1.1 Neuronal Dynamics Comparison

Figure 3 compares SiLIF, AdLIF, and OscSpikeSSM under the same input sequence and firing threshold. SiLIF exhibits dense reset-driven firing during sustained stimulation, whereas AdLIF reduces repeated firing through its adaptive dynamics (Fabre et al., 2025; Yin et al., 2021). In contrast, OscSpikeSSM maintains a bounded second-order oscillatory trajectory and produces substantially fewer, more temporally separated spikes under the same input and threshold. The distinction is particularly visible during the oscillatory input interval, where the state follows a regular oscillatory evolution while only isolated threshold crossings generate spikes. Following removal of the input, the state smoothly relaxes toward equilibrium without sustained firing, illustrating the stable underdamped dynamics imposed by the proposed parameterisation. Thus, rather than encoding temporal structure primarily through first-order membrane decay or adaptation, OscSpikeSSM retains it in

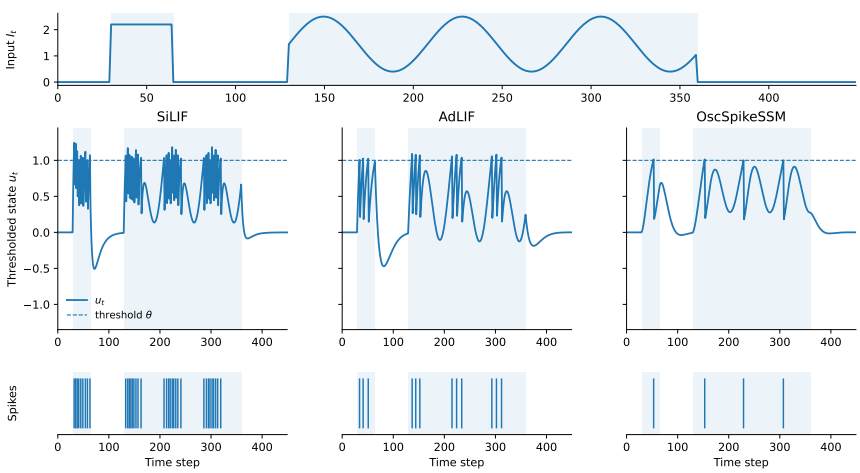

Figure 3: Comparison of SiLIF, AdLIF, and OscSpikeSSM. OscSpikeSSM exhibits sparser spike activity together with smooth second-order oscillatory latent-state evolution.

the amplitude and phase evolution of its coupled oscillatory states while using sparse spikes as the inter-layer communication mechanism.

A.1.2 Damped-IMEX Discretization

Following IMEX discretisations of second-order oscillatory dynamics (Rusch & Rus, 2025), we discretise the continuous state equations of the proposed model using a damped implicit-explicit update. At timestep t, the discrete states $u_t, v_t \in \mathbb{R}^h$, the projected input $I_t \in \mathbb{R}^h$, the spike vector $s_t \in \{0,1\}^h$, and the effective force $F_t \in \mathbb{R}^h$ are defined over the hidden dimension. The non-damping terms are collected into the effective force F_t, which is then used in the damped velocity update as

$$F_t = -k\,u_{t-1} + B\,I_t - R\,s_{t-1}, \qquad v_t = \frac{v_{t-1} + \Delta t\,F_t}{1 + \Delta t\,c}. \tag{1}$$

The damping term is treated implicitly through the denominator, while the restoring force, input drive, and spike-dependent feedback are evaluated from the previous state. A spike produces a subtractive reset of the position-like state, $\widetilde{u}_{t-1} = u_{t-1} - r_{\text{reset}} s_{t-1}$, where $\widetilde{u}_{t-1} \in \mathbb{R}^h$, after which the state is evolved as $u_t = \widetilde{u}_{t-1} + \Delta t\,v_t$. The spike-dependent feedback and subtractive reset act on the dynamics in different ways: the former changes the oscillator's future trajectory by altering the velocity update through R, whereas the latter directly reduces the position-like state after a firing event.

A.1.3 Approximate Parallel Scan

Because each OscSpikeSSM update depends on the previous state and spike, thresholding and reset make the recurrence nonlinear and prevent direct use of the associative scan employed by linear SSMs. Following Newton-based parallelisation of nonlinear recurrences (Danieli et al., 2026; Gonzalez et al., 2024), we formulate the full trajectory as a nonlinear system and linearise it at each Newton iteration. The resulting linearised affine recurrence is then evaluated across the sequence using an associative parallel prefix scan.

Nonlinear sequence formulation: For each hidden mode, we collect the two recurrent states as $\mathbf{z}_t = [u_t, v_t]^\top \in \mathbb{R}^2$. Since the hidden modes are independent under the element-wise OscSpikeSSM dynamics, the following derivation is applied in parallel over all h hidden modes. The recurrence in Section 3.1 can therefore be written as

$$\mathbf{z}_t = f_t(\mathbf{z}_{t-1}, I_t), \qquad \mathbf{z}_0 = \mathbf{0}, \tag{2}$$

where $f_t(\cdot)$ includes the oscillatory state update together with spike-dependent feedback, thresholding, and subtractive reset. Rather than evaluating Equation 2 sequentially, we solve for the complete

trajectory $\mathbf{z}_{1:T}$ simultaneously. At Newton iteration n, the residual associated with timestep t is

$$\mathbf{r}_t^{(n)} = f_t(\mathbf{z}_{t-1}^{(n)}, I_t) - \mathbf{z}_t^{(n)}, \qquad \mathbf{r}_t^{(n)} \in \mathbb{R}^2. \tag{3}$$

The sequential trajectory is recovered when these residuals vanish over the complete sequence. Owing to the first-order dependence on the preceding state, the Jacobian of the stacked residual system is block lower-bidiagonal in time (Danieli et al., 2026; Gonzalez et al., 2024). We therefore avoid constructing the global Jacobian explicitly and operate instead on the local 2×2 transition Jacobians at each timestep.

Local Newton linearization. Let $\Delta\mathbf{z}_t^{(n)}$ denote the Newton correction at timestep t. Linearising the transition around the current trajectory gives

$$\mathbf{J}_t^{(n)} = \left.\frac{\partial f_t(\mathbf{z}, I_t)}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_{t-1}^{(n)}} \in \mathbb{R}^{2\times 2}, \qquad \Delta\mathbf{z}_t^{(n)} = \mathbf{J}_t^{(n)}\Delta\mathbf{z}_{t-1}^{(n)} + \mathbf{r}_t^{(n)}. \tag{4}$$

Thus, each Newton step reduces the nonlinear recurrence to an affine recurrence over the two-dimensional state corrections.

For OscSpikeSSM, let $d = 1 + \Delta t\, c$ and let g_t denote the selected surrogate derivative of the spike function evaluated at $u_{t-1} - \theta$. Suppressing the Newton-iteration superscript for clarity, the local transition Jacobian can be expressed as

$$\mathbf{J}_t = \begin{bmatrix} J_{uu} & J_{uv} \\ J_{vu} & J_{vv} \end{bmatrix}, J_{vu} = -\frac{\Delta t\,(k + R\,g_t)}{d}, \quad J_{vv} = \frac{1}{d}, \quad J_{uu} = 1 - r_{\text{reset}} g_t + \Delta t\, J_{vu}, \quad J_{uv} = \frac{\Delta t}{d}. \tag{5}$$

The forward transition retains hard spike generation, while g_t is used only in the local Jacobian for Newton linearisation. The Jacobian surrogate can therefore be changed without modifying the forward recurrence.

Associative block scan: The affine recurrence in Equation 4 can be written as $\Delta\mathbf{z}_t = \mathbf{A}_t\Delta\mathbf{z}_{t-1} + \mathbf{b}_t$, with $\mathbf{A}_t = \mathbf{J}_t$ and $\mathbf{b}_t = \mathbf{r}_t$. Two consecutive affine transitions can be composed as

$$(\mathbf{A}_2, \mathbf{b}_2) \cdot (\mathbf{A}_1, \mathbf{b}_1) = (\mathbf{A}_2\mathbf{A}_1,\ \mathbf{A}_2\mathbf{b}_1 + \mathbf{b}_2)\,. \tag{6}$$

This operator is associative, which allows the complete correction sequence to be obtained with a parallel prefix scan rather than a sequential forward scan. We use an inclusive Hillis-Steele scan (Hillis & Jr., 1986), where stage j composes transitions separated by 2^j timesteps, for $j = 0, \ldots, \lceil \log_2 T \rceil - 1$. This reduces the temporal dependency depth from T sequential updates to $\mathcal{O}(\log T)$ scan stages per Newton iteration. Since each hidden mode contributes only a 2×2 Jacobian block, the scan is vectorised across batch and hidden dimensions without forming a dense $2h \times 2h$ Jacobian.

Parallel initialisation and Newton update: To initialise the complete trajectory without a sequential rollout, we follow the parallel initialisation strategy used in Newton-based recurrent solvers (Danieli et al., 2026) and evaluate the transition at every timestep from a zero previous state,

$$\mathbf{z}_t^{(0)} = f_t(\mathbf{0}, I_t), \qquad t = 1, \ldots, T. \tag{7}$$

All initial states can therefore be computed simultaneously. The implementation also supports an all-zero initial trajectory. Once the parallel scan produces $\Delta\mathbf{z}_{1:T}^{(n)}$, the trajectory is updated as

$$\mathbf{z}^{(n+1)} = \mathbf{z}^{(n)} + \omega\Delta\mathbf{z}^{(n)}, \tag{8}$$

where ω controls the Newton step magnitude. During the iterations, we track the recurrence residual, the magnitude of the state correction, and the fraction of spikes that change between successive estimates. The solver can terminate after a prescribed number of iterations or once both the residual and the spike pattern have sufficiently stabilised.

After the final Newton iteration, spikes are obtained by applying the same hard threshold as in the sequential OscSpikeSSM to the estimated position-like states. Because only a finite number of Newton iterations is used and the non-differentiable spike threshold is locally approximated by a

Table 8: Inference comparison between sequential recurrence and approximate parallel scan on two UEA datasets - EthanolConcentration and MotorImagery. Inference latency is reported in milliseconds per batch, and speedup is computed relative to sequential execution for each dataset.

Execution	EthanolConcentration			MotorImagery		
	Accuracy (%)	**Inference (ms/batch)**	**Speedup (×)**	**Accuracy (%)**	**Inference (ms/batch)**	**Speedup (×)**
Sequential	29.8 ± 1.0	220.7	1.00	58.3 ± 0.9	370.8	1.00
Approx. parallel ($K = 2$)	28.8 ± 1.6	134.5	$\uparrow 1.6\times$	58.2 ± 3.6	257.4	$\uparrow 1.4\times$

surrogate derivative, the recovered trajectory may differ from the exact sequential solution (Danieli et al., 2026; Gonzalez et al., 2024; Neftci et al., 2019). During training, gradients are computed via reverse-mode automatic differentiation using unrolled Newton iterations and parallel prefix scans; neither implicit differentiation nor BPTT via the exact sequential recurrence is used. For the reported experiments, the Newton Jacobian uses a boxcar spike derivative, which is piecewise constant with respect to the state, so no second-order surrogate derivative is required. Gradients propagate through the residual evaluation, affine scan, Newton updates, and model parameters, while hard spike generation retains the standard first-order surrogate-gradient backward. We therefore refer to this execution path as an approximate parallel scan and use the sequential recurrence as the reference trajectory.

For the approximate parallel scan, we use $K = 2$ Newton iterations with damping factor $\omega = 1.0$, initialising the trajectory by evaluating each timestep in parallel from a zero previous state. Accuracy is reported as mean $\pm$ standard deviation over five runs. Inference latency is measured using the same architecture, batch size, and hardware for both execution modes, with CUDA synchronisation and warm-up iterations before timing. As shown in Table 8, the approximate parallel scan reduces inference latency from 220.7 to 134.5 ms/batch on Ethanol and from 370.8 to 257.4 ms/batch on Motor, corresponding to $1.6\times$ and $1.4\times$ speedups, respectively, while maintaining comparable classification accuracy. These results indicate that reducing the sequential dependence of the recurrence provides a clear wall-clock advantage without substantially degrading task performance. This acceleration comes with a memory trade-off, as the parallel formulation requires higher peak memory to store the sequence-wide intermediate quantities and Jacobian terms used by the boxcar-based Newton scan.

A.1.4 Role of Spike Reset

The second-order dynamics of OscSpikeSSM are designed to retain temporal information through a persistent oscillatory state. After a threshold crossing, however, oscillatory spiking dynamics can lead to repeated firing when the state remains close to the firing boundary, a behaviour previously observed in resonate-and-fire neurons (Higuchi et al., 2024). We therefore apply the subtractive reset introduced in Sec. 3.1, consistent with commonly used post-spike membrane-reset mechanisms in recurrent SNNs (Bellec et al., 2018), where $\tilde{u}_{t-1} - \theta = (u_{t-1} - \theta) - r_{\text{reset}}$ for $s_{t-1} = 1$. This operation directly increases the post-spike distance from the firing threshold, with r_{reset} controlling subsequent excitability: smaller values permit faster re-firing, whereas larger values require stronger state evolution to cross the threshold again.

Unlike a hard reset, the subtractive reset does not force the latent state to a fixed value. Residual information in the position-like state is preserved, while the velocity-like state v_t is not directly reset, allowing the second-order trajectory to continue across spike events. This discourages redundant spike generation and promotes sparser activity, analogous to refractory regulation in biological spiking neurons (Higuchi et al., 2024). Since synaptic operation counts in SNNs scale with spike activity, reduced firing can lower the number of SOPs and the associated operation-based energy cost. The reset and spike-dependent feedback act on different components of the dynamics: the reset modifies the post-spike position state, whereas the feedback term alters the subsequent velocity update, thereby affecting the future oscillatory trajectory. Together, they regulate spiking while preserving the underlying oscillator dynamics.

A.1.5 THEORETICAL ENERGY ESTIMATION

We estimate theoretical compute energy using an operation-level proxy commonly adopted for ANN and SNN efficiency analysis (Horowitz, 2014). For ANN models, computation is treated as dense multiply-accumulate (MAC) operations, giving

$$E_{\mathrm{ANN}} = E_{\mathrm{MAC}} \sum_{\ell} \mathrm{FLOPs}_{\ell}. \tag{9}$$

For SNN models, the real-valued input layer is computed densely using MACs, while subsequent spike-driven layers are modelled using accumulate (AC) operations. The synaptic operation count of layer ℓ is estimated as

$$\mathrm{SOPs}_{\ell} = r_{\ell-1}\,\mathrm{FLOPs}_{\ell}, \tag{10}$$

where $r_{\ell-1}$ is the average presynaptic firing rate entering that layer. The corresponding SNN energy estimate is

$$E_{\mathrm{SNN}} = E_{\mathrm{MAC}}\,\mathrm{FLOPs}_{1} + E_{\mathrm{AC}} \sum_{\ell=2}^{N_{\mathrm{lyr}}} \mathrm{SOPs}_{\ell}. \tag{11}$$

Following the 45 nm CMOS estimates of Horowitz (2014), we use $E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$ and $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$ per operation. Reported values are therefore analytical compute-energy estimates based on operation counts and measured firing activity, rather than direct hardware power measurements.

A.2 EXTENDED RESULTS

A.2.1 COMPUTATIONAL EFFICIENCY METRICS

Unless stated otherwise, model memory denotes the static FP32 storage required by the learnable model parameters. Theoretical compute energy is estimated from the numbers of multiply-accumulate (MAC) and accumulate (AC) operations using the operation-level energy model (Appendix A.1.5) across all compared methods; these values therefore represent analytical compute-energy estimates. Wall-clock inference is measured on an NVIDIA RTX 4500 Ada Generation GPU at batch size 1. Table 2 reports full-test-set inference time averaged over three runs; Table 3 reports per-sample latency over 100 timed iterations; Table 5 reports per-sample latency over 1000 timed iterations; Table 6 reports per-clip latency as the mean and standard deviation over three runs of 100 timed iterations; and Table 7 reports per-sample batch-1 inference latency.

A.2.2 LONG RANGE CLASSIFICATION

We consider the three long sequence datasets - EigenWorms, EthanolConcentration, and MotorImagery from the UEA multivariate time-series archive, spanning sequence lengths from moderately long trajectories to the 17,984-step EigenWorms benchmark. This experiment tests the long-range sequence modelling capability of the proposed second-order spiking dynamics (OscSpikeSSM) before evaluating the same core formulation on the more specialised radar-based tasks that follow. Although recent resonate-and-fire (R&F) neuron-based SSMs have achieved strong performance on long-range classification benchmarks (Agrawal et al., 2026), their hardware realisation can require larger circuit footprints and silicon area than simpler LIF-family neurons (Le et al., 2023). This can reduce the achievable neuron density on area-constrained hardware, limiting the number of neurons that can be implemented within a given chip area. This hardware gap motivates the development of oscillatory spiking dynamics that retain the simplicity of LIF-family neuron implementations while improving their ability to model long temporal dependencies. Table 9 provides a comprehensive comparison of OscSpkSSM with transformer, first-order SSM, second-order SSM, and spiking baselines across the three long-range classification datasets.

A.2.3 MMWAVE SENSING WITH OSCSPIKESSM

Each radar frame retains up to 64 points represented by (x, y, z, v) and is spatially encoded before temporal modelling. The following sections provide the dataset-specific spatial representations, training configurations, and evaluation settings for AOPHand, mmFiT, and MMActivity.

Architecture

Table 9: Classification accuracy (%) on selected long-range UEA multivariate time-series benchmarks. Results are reported as the mean ± standard deviation across five training runs with different seeds. Sequence length and number of classes are shown below each dataset name. †OscSpkSSM does not employ auxiliary nonlinear activation functions such as GELU or GLU.

Seq/Class		17,984/5	1,751/4	3,000/2	
	LIF-SNN	**Worms**	**Ethanol**	**Motor**	**Avg**
Transformer Methods					
Transformer (2017)	No	OOM	40.5 ± 6.3	50.5 ± 3.0	45.5
RFormer (2024)	No	90.3 ± 0.1	34.7 ± 4.1	55.8 ± 6.6	60.3
First-Order Methods					
NCDE (2020)	No	75.0 ± 3.9	29.9 ± 6.5	49.5 ± 2.8	51.5
NRDE (2021)	No	83.9 ± 7.3	25.3 ± 1.8	47.0 ± 5.7	52.1
Log-NCDE (2024)	No	85.6 ± 5.1	34.4 ± 6.4	53.7 ± 5.3	57.9
S5 (2023)	No	81.1 ± 3.7	24.1 ± 4.3	47.7 ± 5.5	51.0
LRU (2023)	No	87.8 ± 2.8	21.5 ± 2.1	48.4 ± 5.0	52.6
S6 (2024)	No	85.0 ± 16.1	26.4 ± 6.4	51.3 ± 4.7	54.2
Mamba (2024)	No	70.9 ± 15.8	27.9 ± 4.5	47.7 ± 4.5	48.8
AUSSM* (2025)	No	82.6 ± 3.4	4.7 ± 4.1	2.6 ± 5.5	30.0
PD-SSM (2025)	No	90.0 ± 5.7	34.7 ± 4.0	60.0 ± 3.7	61.6
Second-Order SSM Methods					
BioOSS* (2025)	No	92.8 ± 5.2	33.4 ± 10.7	55.8 ± 5.8	60.7
LinOSS (2025)	No	95.0 ± 4.4	29.9 ± 0.6	60.0 ± 7.5	61.6
Lin-HSSM-BPTT (2025)	No	78.3 ± 7.5	29.9 ± 0.6	48.1 ± 5.7	52.1
Lin-HSSM-RHEL (2025)	No	75.0 ± 9.9	29.9 ± 0.6	51.6 ± 5.0	52.2
Nonlin-HSSM-BPTT (2025)	No	51.1 ± 7.2	29.9 ± 0.6	56.5 ± 7.6	45.8
Nonlin-HSSM-RHEL (2025)	No	50.6 ± 6.7	29.9 ± 0.6	53.0 ± 5.7	44.5
Spiking Methods					
SiLIF/AdLIF (2025)	Yes	78.5 ± 6.1	29.6 ± 1.1	55.6 ± 2.9	54.6
P-SpikeSSM (2025)	Yes	85.5 ± 7.2	32.2 ± 6.1	51.3 ± 7.5	56.3
SpikingSSM (2025)	Yes	85.6 ± 2.3	21.6 ± 7.4	55.6 ± 12.4	54.3
OscSpkSSM (Ours)†	**Yes**	$\mathbf{85.4 \pm 1.9}$	$\mathbf{29.8 \pm 1.0}$	$\mathbf{58.3 \pm 0.9}$	**57.8**

Point-Cloud Sequence Representation: Given the point-cloud representation introduced in the main text, longer or denser radar frames are uniformly sub-sampled, while shorter inputs are zero-padded. This yields $\mathbf{X} \in \mathbb{R}^{B\times T\times P\times 4}$, which is flattened over the point and feature dimensions to $\mathbf{X} \in \mathbb{R}^{B\times 4\times 256}$. AOPHand and mmFiT are normalised using training-set channel statistics, whereas MMActivity retains its native feature scale.

Frame-Level Spatial Encoding: In mm-OscSpikeSSM, each of the four radar frames is independently processed by a shared convolutional encoder, with weights reused across time to extract compact spatial features while preserving frame order. The encoder uses three convolutional layers, global spatial pooling, and a linear projection, producing $\mathbf{F} \in \mathbb{R}^{B\times T\times D}$. The resulting sequence is then passed to the OscSpikeSSM backbone, while the spatial arrangement is adapted to each dataset.

AOPHand: For AOPHand, the 256-dimensional frame vector is reshaped to a single-channel 16×16 map. The spatial encoder uses convolutional widths of 64 and 96 with SiLU activations and projects each frame to a 256-dimensional embedding. The resulting four-frame sequence is processed by one bidirectional OscSpikeSSM layer with 64 hidden units per temporal direction. The forward and backward representations are concatenated before deterministic temporal averaging and classification into the six gesture classes.

mmFiT: For mmFiT, the point structure is retained more explicitly by interpreting the four point attributes (x, y, z, v) as channels and arranging the 64 points on an 8×8 grid. The resulting $4\times8\times8$ frame representation is processed by the spatial encoder with channel widths 64 and 128 and SiLU activations, producing a 128-dimensional frame embedding. Temporal modelling remains causal and is performed by two stacked unidirectional OscSpikeSSM layers with 64 hidden units each. The two layers are followed by mean temporal pooling and classification into the seven fitness activities.

MMActivity: MMActivity uses the single-channel 16×16 frame representation, as in AOPHand, but employs a smaller spatial and temporal model to account for the limited dataset size. The spatial encoder uses convolutional widths of 48 and 96 with GELU activations and projects each frame to a 96-dimensional embedding. A single bidirectional OscSpikeSSM layer with 32 hidden units per direction then models the four-frame sequence. The forward and reverse temporal features are concatenated and averaged across time before the final linear classifier predicts one of the five activity classes.

Spiking State-Space Processing and Readout: Let $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_T] \in \mathbb{R}^{B \times T \times D}$ denote the sequence of frame-level embeddings produced by the spatial encoder. At each time step, $\mathbf{f}_t$ drives the OscSpikeSSM dynamics, which update the continuous-valued oscillatory states and generate a thresholded spike representation for temporal propagation. After the final OscSpikeSSM layer, the sequence representation is averaged over the T frames and passed directly to a linear classification head. No additional recurrent or attention-based temporal module is applied after the OscSpikeSSM backbone.

Taken together, these task-specific configurations retain a common mm-OscSpikeSSM design while adapting the spatial representation and temporal organisation to each dataset. Relative to the reference SpikingLeNet pipeline (Liao et al., 2026), the temporal component replaces first-order membrane-based integration with second-order oscillatory state-space dynamics over the radar embedding sequence. The same OscSpikeSSM core therefore supports both bidirectional context aggregation and causal temporal modelling while preserving sparse spike-based communication.

Evaluation Settings

Evaluation Protocol: We follow the $80/20$ train-test partition produced by the point-cloud preprocessing pipeline for all three datasets (Liao et al., 2026). AOPHand contains $2017/504$ train/test samples across six classes, mmFiT $2475/619$ samples across seven classes, and MMActivity $160/40$ samples across five classes. Since no official validation split is provided, we use the provided test split for checkpoint selection and report the accuracy of the best checkpoint selected on this split; no separate held-out test evaluation is performed.

Training Configuration: All models are optimised using AdamW with dataset-specific configurations. AOPHand is trained for 200 epochs with batch size 64, learning rate 1.45×10^{-3}, weight decay 1.72×10^{-3}, threshold 0.60, reset magnitude 1.0, gradient clipping at 2.0, and spike-activity regularisation weight 1.39×10^{-3}.

mmFiT is trained for 150 epochs with batch size 16, learning rate 1.55×10^{-3}, weight decay 9.84×10^{-5}, threshold 0.55, reset magnitude 0.75, gradient clipping at 1.0, and spike-activity regularisation weight 2.49×10^{-3}.

MMActivity is trained for 150 epochs with batch size 8, learning rate 2.00×10^{-3}, weight decay 1.89×10^{-5}, threshold 0.25, reset magnitude 1.0, and gradient clipping at 1.0. Owing to its smaller training set, we use label smoothing of 0.05, disable class weighting, and apply a firing-rate regulariser that encourages activity within $[0.02, 0.10]$.

A.2.4 Performance on Google Soli

Architecture

Range-Doppler Sequence Representation: Each Soli sample contains $T = 40$ range-Doppler frames. Following the established Soli setting (Yin et al., 2021), a single radar channel is retained and each frame is flattened into a feature vector $\mathbf{x}_t \in \mathbb{R}^F$, yielding $\mathbf{X} \in \mathbb{R}^{B \times T \times F}$ with $F = 1024$. The range-Doppler sequence is passed directly to the temporal encoder without an additional feature extractor.

Temporal Spiking State-Space Encoder: The sequence is processed causally by two stacked OscSpikeSSM layers, each with 256 hidden units. At frame t, the representation therefore depends only on observations available up to that frame, while the oscillatory state dynamics accumulate temporal information over the gesture sequence. This retains the streaming nature of the Soli task while also preserving the temporal information over the course of the gesture.

Table 10: Classification accuracy and parameter count on AOPHand, mmFiT, and MMActivity datasets. Reference results are reported as mean ± standard deviation over three runs, while our results are reported over five runs with different seeds. Text in bold and underlined denotes the first- and second-best accuracies and the smallest and second-smallest SNN models.

Baseline	AOPHand		mmFiT		MMActivity		Avg
	Accuracy (%)	Params (M)	Accuracy (%)	Params (M)	Accuracy (%)	Params (M)	
MLP	$69.70_{\pm 0.99}$	4.327	$66.51_{\pm 2.02}$	4.327	$60.83_{\pm 2.89}$	4.327	65.68
LeNet	$60.86_{\pm 1.68}$	4.191	$62.36_{\pm 0.58}$	4.192	$59.17_{\pm 3.82}$	4.191	60.80
VGG9	$74.39_{\pm 0.61}$	31.603	$69.36_{\pm 2.15}$	31.605	$70.00_{\pm 4.33}$	31.601	71.25
VGG16	$67.92_{\pm 6.02}$	70.313	$65.43_{\pm 0.58}$	70.315	$62.50_{\pm 4.33}$	70.311	65.28
ResNet18	$72.47_{\pm 1.30}$	11.173	$71.94_{\pm 1.29}$	11.174	$61.67_{\pm 6.29}$	11.173	68.69
ResNet50	$72.54_{\pm 0.80}$	23.514	$71.84_{\pm 1.97}$	23.516	$61.67_{\pm 2.89}$	23.512	68.68
ResNet101	$69.70_{\pm 0.72}$	42.506	$72.64_{\pm 0.25}$	42.508	$64.17_{\pm 1.44}$	42.504	68.84
RNN	$26.40_{\pm 9.25}$	0.026	$22.56_{\pm 5.93}$	0.026	$48.33_{\pm 1.44}$	0.025	32.43
GRU	$67.52_{\pm 5.73}$	0.075	$14.11_{\pm 0.38}$	0.075	$47.50_{\pm 2.50}$	0.075	43.04
LSTM	$20.71_{\pm 1.12}$	0.100	$13.14_{\pm 0.52}$	0.100	$54.17_{\pm 7.64}$	0.100	29.34
BiLSTM	$46.14_{\pm 1.40}$	0.200	$12.98_{\pm 0.89}$	0.200	$45.83_{\pm 3.82}$	0.200	34.98
CNN-GRU	$61.98_{\pm 9.44}$	0.463	$67.80_{\pm 0.49}$	0.463	$65.00_{\pm 2.50}$	0.463	64.93
ViT	$21.39_{\pm 3.31}$	2.176	$36.40_{\pm 5.65}$	2.176	$65.83_{\pm 5.77}$	2.175	41.21
SpikingLeNet	$83.70_{\pm 4.24}$	4.191	$73.67_{\pm 1.55}$	4.192	$75.00_{\pm 6.61}$	4.191	77.46
SpikingVGG9	$88.51_{\pm 0.40}$	31.603	$75.93_{\pm 1.82}$	31.605	$\underline{75.83_{\pm 6.29}}$	31.601	80.09
SpikingVGG16	$86.86_{\pm 0.60}$	70.313	$70.65_{\pm 2.51}$	70.315	$65.83_{\pm 2.89}$	70.311	74.45
SEW-ResNet18	$88.97_{\pm 0.64}$	11.173	$78.57_{\pm 1.21}$	11.174	$66.67_{\pm 3.82}$	11.173	78.07
SEW-ResNet50	$86.01_{\pm 1.02}$	23.514	$76.20_{\pm 1.17}$	23.516	$65.83_{\pm 1.44}$	23.512	76.01
MS-ResNet18	$90.96_{\pm 0.31}$	11.178	$85.08_{\pm 0.73}$	11.178	$65.83_{\pm 3.82}$	11.177	80.62
MS-ResNet50	$89.51_{\pm 0.69}$	23.514	$85.41_{\pm 0.83}$	23.516	$67.50_{\pm 4.33}$	23.512	80.81
SpikFormer	$91.29_{\pm 1.58}$	2.566	$83.58_{\pm 0.52}$	2.567	$\underline{75.83_{\pm 1.44}}$	2.566	$\underline{83.57}$
SDT	$88.31_{\pm 0.52}$	6.081	$82.87_{\pm 1.32}$	6.081	$69.17_{\pm 3.82}$	6.080	80.12
QKFormer	$90.17_{\pm 0.58}$	$\underline{2.414}$	$83.36_{\pm 0.70}$	$\underline{2.414}$	$66.67_{\pm 2.89}$	$\underline{2.414}$	80.07
SpikingResFormer	$\mathbf{92.74_{\pm 0.64}}$	17.310	$\underline{86.38_{\pm 0.74}}$	17.310	$65.83_{\pm 3.82}$	17.309	81.65
mm-OscSpikeSSM	$\underline{92.14_{\pm 0.32}}$	$\mathbf{0.182}_{\downarrow 95.11\times}$	$\mathbf{90.02_{\pm 0.52}}$	$\mathbf{0.255}_{\downarrow 67.88\times}$	$\mathbf{78.00_{\pm 1.87}}$	$\mathbf{0.138}_{\downarrow 125.42\times}$	**86.72**

Temporal Classification Readout: The output of the final OscSpikeSSM layer is passed to a non-spiking leaky-integrator readout. For hidden representation $\mathbf{h}_t$, the readout evolves as

$$\mathbf{r}_t = \mathbf{W}_o \mathbf{h}_t + \mathbf{b}_o, \qquad \mathbf{u}_t = \boldsymbol{\alpha} \mathbf{u}_{t-1} + (1 - \boldsymbol{\alpha}) \mathbf{r}_t, \tag{12}$$

where $\boldsymbol{\alpha}$ is a learnable decay parameter and $\mathbf{u}_t$ provides the class logits at frame t. For whole-sequence classification, logits from the final eight frames are averaged to obtain the sequence-level prediction. The same output trajectory is also used for frame-level and early-recognition evaluation.

Relative to the recurrent SRNN formulation (Yin et al., 2021), which uses recurrently connected LIF/ALIF neurons with adaptive membrane dynamics, the temporal encoder replaces first-order recurrent integration with second-order oscillatory state-space evolution. The resulting states capture decaying oscillatory responses across successive radar frames, providing a temporal mechanism suited to dynamic gesture motion while sparse spikes serve as the inter-layer representation. Unlike the reference Soli implementation, which trains separate networks for streaming and whole-sequence recognition, our temporal readout produces frame-wise logits throughout the gesture and uses the same output trajectory for the final sequence prediction, enabling both settings with a single trained model.

Evaluation

Evaluation Protocol: We use the official Google Soli train-test split containing 1,386 training and 1,364 test sequences across 11 gesture classes (Lien et al., 2016). Since no separate validation split is used, the provided test split is employed for checkpoint selection. For each run, the checkpoint with

Table 11: Gesture recognition performance on the Google Soli dataset. Sequence accuracy measures complete-gesture classification, while frame accuracy measures recognition at individual timesteps. Inference efficiency comparison on the Soli dataset. Energy denotes the theoretical compute energy per sample, while latency is the measured batch-1 wall-clock inference time on GPU.

Model	Sequence Acc. (%)
CNN	77.7
CNN+LSTM	87.2
SRNN	91.9
Ours	96.80 ± 0.27

Model	Frame Acc. (%)	Memory (MiB) ↓	Energy (μJ) ↓	Latency (ms) ↓
LSTM	77.7	20.053	965.726	4.479
ReLU	79.6	4.027	193.974	5.404
SRNN	79.8	4.035	97.491	13.663
Ours	78.36 ± 0.51	1.274	48.409	8.568

Table 12: Early gesture-recognition performance on Google Soli as progressively more radar frames are observed. Results are reported as mean $\pm$ standard deviation over five runs.

Frames Observed	10	20	30	40
Accuracy (%)	62.00 ± 0.49	83.48 ± 1.07	94.28 ± 0.43	95.09 ± 0.40

the highest sequence-level validation accuracy is selected and its accuracy is reported; no separate held-out test evaluation is performed.

Training Configuration: The training settings are selected to get the best possible performance using an Optuna hyperparameter search and then fixed for the final experiments. The model is trained for 150 epochs using AdamW with a batch size of 8, learning rate 2.82×10^{-3}, and weight decay 3.00×10^{-5}. The spiking threshold and reset amount are set to 0.15 and 0.10, respectively, and gradients are clipped to a maximum norm of 2.0. The training objective combines sequence-level and frame-level classification losses, both weighted by 1.0, together with a spike-rate regularisation term weighted by 0.03.

Reference Models and Performance Metrics: We compare against the CNN and CNN-LSTM baselines from the original Soli study (Lien et al., 2016) and the adaptive SRNN (Yin et al., 2021). Sequence accuracy is used for complete-gesture classification, while frame-level accuracy characterises recognition during the gesture. Early-recognition performance is evaluated after observing 10, 20, 30, and 40 frames to measure how prediction accuracy evolves as additional radar observations become available.

Results: The sequence-level and computational-efficiency comparison with SRNN is discussed in the main text. Table 11 additionally shows that the proposed model achieves higher sequence accuracy than the CNN (77.7%) and CNN-LSTM (87.2%) baselines. At the frame level, it achieves $78.36 \pm 0.51\%$, outperforming LSTM (77.7%) and remaining close to ReLU (79.6%) and SRNN (79.8%). The early-recognition results in Table 12 further show that accuracy increases from $62.00 \pm 0.49\%$ at 10 frames to $83.48 \pm 1.07\%$ at 20 frames and $94.28 \pm 0.43\%$ at 30 frames, reaching $95.09 \pm 0.40\%$ at 40 frames. This indicates that most gesture-discriminative information is captured before the complete sequence is observed.

A.2.5 Radar based 3D Pose Estimation using Spiking SSM

MMVR-OscSpikeSSM Architecture: Given T consecutive horizontal-depth and vertical-depth radar heatmaps, $\mathbf{X}_{\text{hor}} \in \mathbb{R}^{B \times T \times W \times D}$ and $\mathbf{X}_{\text{ver}} \in \mathbb{R}^{B \times T \times H \times D}$, the two views are independently projected using dual-view convolutional patch embeddings while preserving the temporal dimension. The resulting features, $\mathbf{F}_{\text{hor}}^{(0)} = \mathcal{E}_{\text{hor}}(\mathbf{X}_{\text{hor}})$ and $\mathbf{F}_{\text{ver}}^{(0)} = \mathcal{E}_{\text{ver}}(\mathbf{X}_{\text{ver}})$, pass through a two-stage encoder, where each view is processed by RF-SpikingOSS blocks and then coupled through depth-aware cross-view fusion. The second stage operates at reduced spatial and depth resolution before the fused features are passed to the pose head.

RF-SpikingOSS Block: Given an input feature $\mathbf{F} \in \mathbb{R}^{B\times T\times S\times D\times C}$, where S, D, and C denote the spatial, depth, and channel dimensions, respectively, OscSpikeSSM processes the normalised input through two parallel branches. The heatmap branch consists of HM-SpkScan followed by HM-SFFN, whereas the spatio-temporal branch uses SpikingSTScan (Figure 2) to model the feature tensor along its spatial and temporal dimensions. Their outputs are independently gated, concatenated, projected back to the original channel dimension, and combined with a residual connection. Both branches use an omnidirectional spiking scan along the principal radar axes: temporal, spatial, and depth, $\mathcal{D} = \{T^+, T^-, S^+, S^-, D^+, D^-\}$, where $S = W$ for the horizontal view and $S = H$ for the vertical view. Each directional sequence is processed by a shared OscSpikeSSM core, after which the six outputs are concatenated and projected to recover the original feature dimension. This preserves the structured radar geometry while enabling information to propagate along each of its principal axes.

Depth-Aware Cross-View Fusion: Given $\mathbf{F}_{\text{hor}} \in \mathbb{R}^{B\times T\times W'\times D'\times C}$ and $\mathbf{F}_{\text{ver}} \in \mathbb{R}^{B\times T\times H'\times D'\times C}$, the shared depth dimension is used for cross-view interaction. Each stream is spatially aggregated to obtain depth-aligned features, which are combined into a shared depth context, then projected back into the horizontal and vertical streams and added residually. This exchanges information across views while preserving their view-specific spatial structure.

Weakly-Supervised Pose Decoding and Training: The final horizontal and vertical features are globally aggregated into a shared radar context and combined with N_q learnable pose queries. Each query predicts an initial $J = 17$ joint pose, $\widetilde{\mathbf{P}}_i = \sigma(\mathcal{H}_{\text{init}}(\mathbf{q}_i))$, which is refined as $\widehat{\mathbf{P}}_i = \sigma\Big(\sigma^{-1}(\widetilde{\mathbf{P}}_i) + \mathcal{H}_{\text{ref}}(\mathbf{q}_i)\Big)$, where $\sigma(\cdot)$ denotes the element-wise sigmoid function and $\sigma^{-1}(\cdot)$ its inverse logit transform, with a separate head assigning a confidence score to each query, and the refined prediction is used as the final 3D pose estimate. Training does not use fine-grained 3D keypoint supervision. Instead, the query whose predicted pose centroid is closest to the target 3D bounding-box centroid is selected, $q^* = \arg\min_q \|\mathbf{c}_q - \mathbf{c}_{\text{GT}}\|_2$., where $\mathbf{c}_q$ is the centroid of the refined pose predicted by query q and $\mathbf{c}_{\text{GT}}$ is the target centroid. The matched pose is supervised by 3D box and centroid constraints, projected 2D keypoint and object keypoint similarity (OKS) losses, a weak bone-structure constraint, and query-confidence supervision. The overall objective is, $\mathcal{L} = \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{gravity}}\mathcal{L}_{\text{gravity}} + \lambda_{\text{bone}}\mathcal{L}_{\text{bone}} + \lambda_{\text{kpt2D}}\mathcal{L}_{\text{kpt2D}} + \lambda_{\text{OKS}}\mathcal{L}_{\text{OKS}} + \lambda_{\text{conf}}\mathcal{L}_{\text{conf}}$.

Architectural Novelty: Our model addresses the same weakly supervised 3D pose-estimation setting as RAPTR (Kato et al., 2025), but differs in feature modelling, cross-view fusion, and pose decoding. Rather than using convolutional features followed by Transformer-based cross-view interaction and attention-based pose refinement, MMVR-OscSpikeSSM preserves the temporal, spatial, and depth axes of the horizontal and vertical radar views and processes them using multidirectional oscillatory-spiking SSM scans. Cross-view interaction is performed through their shared depth dimension: spatially aggregated, depth-aligned features are combined into a common context and projected residually back into both streams. Pose estimation is then performed from the fused radar representation using learnable pose queries that predict an initial 3D pose and a coordinate refinement, avoiding the deformable-attention pose-joint decoder used by RAPTR. The model also removes RAPTR's predefined 3D pose template; instead, the refined joints determine the enclosing 3D bounding box and centroid, while weak 3D geometric supervision, projected 2D keypoint/OKS losses, and a bone-structure constraint guide pose learning.

Comparison with Baseline Spiking SSM Models: We further adapt P-SpikeSSM (Bal & Sengupta, 2025) to MMVR as MMVR-P-SpikeSSM baseline. Its original sequence backbone is retained, with only certain modifications in input encoding and output regression required for pose estimation. The horizontal and vertical heatmaps are combined into a single representation, and the four-frame input is arranged as a sequence of 1024 tokens with feature dimension 256. P-SpikeSSM retains its probabilistic spiking SSM formulation, using probabilistic spike generation to support parallelizable long-range sequence modelling. In contrast, MMVR-OscSpikeSSM preserves the horizontal and vertical heatmaps as separate structured streams and models their temporal, spatial, and depth dimensions through multidirectional spiking state-space scans. Their shared depth coordinate is used for explicit cross-view fusion, and the resulting representation is decoded using query-based 3D pose prediction and refinement rather than direct regression from a pooled token sequence. The comparison therefore isolates the contribution of radar-structured representation,

axis-aware state-space modelling, depth-guided fusion, and structured pose decoding beyond the use of a general-purpose spiking SSM backbone.

Evaluation

Evaluation Protocol: We evaluate the models on the P1S1 protocol of MMVR, comprising a single subject recorded in an open-space setting, and retain the official train, validation, and test partitions without re-splitting. The resulting split contains $86{,}579$ training, $10{,}538$ validation, and $10{,}785$ test samples. Each sample consists of $T = 4$ synchronised horizontal-depth and vertical-depth radar frames with heatmap resolution 256×128. The radar sequences are used together with the available 2D keypoint and bounding-box annotations for weakly supervised 3D pose estimation. Fine-grained 3D keypoints are not used as training targets.

Training Configuration: MMVR-OscSpikeSSM uses an embedding dimension of 64, two encoder stages with feature dimensions $(64, 64)$ and one block per stage, patch size 4, $N_q = 10$ pose queries, and $J = 17$ body joints. The spiking threshold and reset amount are both 0.25. The model is optimised with AdamW using a learning rate of 2×10^{-4}, weight decay of 10^{-4}, batch size 16, gradient clipping at 0.05, and bfloat16 mixed-precision training. The loss weights are $\lambda_{\text{box}} = 1$, $\lambda_{\text{gravity}} = 1$, $\lambda_{\text{bone}} = 0.05$, $\lambda_{\text{kpt2D}} = 0.001$, $\lambda_{\text{OKS}} = 0.001$, and $\lambda_{\text{conf}} = 0.1$.

Reference Models: We compare MMVR-OscSpikeSSM with RAPTR (Kato et al., 2025) and the spiking SSM baselines MMVR-P-SpikeSSM (Bal & Sengupta, 2025). RAPTR serves as the attention-based reference for weakly supervised 3D pose estimation on MMVR, while MMVR-P-SpikeSSM is evaluated as direct-transfer spiking state-space baselines. Both use the same P1S1 split and four-frame protocol, with batch size 4, learning rate 5×10^{-4}, and weight decay 10^{-2}. To provide a consistent task interface, both are trained using the 3D bounding-box and gravity-centre constraints. These experiments, therefore, measure the direct applicability of the original spiking SSM backbones to MMVR rather than the effect of an MMVR-specific redesign.

Pose Evaluation Measures and Model Selection: Because MMVR does not provide fine-grained 3D keypoint ground truth for evaluation, MPJPE cannot be computed directly. Following the RAPTR evaluation protocol, we construct the enclosing 3D bounding box from the predicted 17 joints and compare it with the target box. We report the Euclidean distance between the predicted and target box centres, as well as the absolute errors in the horizontal, vertical, and depth edge lengths. Normalised radar coordinates are converted to physical units using axis spans of $(5, 5, 8)$ m, and all errors are reported in centimetres.

At the end of each epoch, the model is evaluated on the validation set using the composite score $\mathcal{S}_{\text{val}} = e_{\text{center}} + e_h + e_v + e_d$, where the four terms denote the centre-distance and horizontal, vertical, and depth edge-length errors, respectively. The checkpoint with the lowest validation score is retained. After training, the selected checkpoint is evaluated once on the held-out test split.

A.2.6 Performance on Ego-Centric Hand Pose Estimation

Architecture

Spatially Preserved Temporal Encoding: Given a radar sequence $\mathbf{X}_r \in \mathbb{R}^{B\times T\times C\times H\times W}$, where T, C, H, and W denote the temporal, channel, height, and width dimensions, respectively, a factorized 3D convolutional embedding reduces the spatial resolution while preserving the temporal dimension. The resulting features are projected to $\mathbf{F}_r \in \mathbb{R}^{B\times T\times S\times D}$, with $S = 8 \times 4 = 32$ spatial locations per frame. Learned spatial and temporal embeddings are added before temporal processing.

For each spatial location s, the sequence $\mathbf{F}_{r,:,s} \in \mathbb{R}^{B\times T\times D}$ is processed by a shared bidirectional OscSpikeSSM encoder. The same encoder parameters are reused across all locations, preserving the radar spatial structure while modelling temporal evolution independently at each position. The encoded features are combined with the corresponding input tokens through a residual connection.

Wrist-Guided Spatial Context: A higher-resolution radar feature map is processed by an auxiliary wrist-prior branch to provide explicit hand-localisation cues. A spatial probability distribution is used to estimate a coarse wrist location and a feature descriptor centred on the corresponding hand

region. These descriptors are projected to the dense-token dimension and appended to the frame representation, providing localised hand context while retaining the broader spatial radar features.

IMU-Guided Radar Conditioning: The synchronised IMU sequence $\mathbf{X}_i \in \mathbb{R}^{B\times T\times F}$ is processed by a temporal spiking SSM to capture motion information across frames. The resulting IMU representation conditions the dense radar tokens through feature-wise scaling and shifting,

$$\widetilde{\mathbf{F}}_r = \mathbf{F}_r\,(1+\alpha\boldsymbol{\gamma}) + \alpha\boldsymbol{\beta},$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are predicted from the IMU features and α controls the modulation strength. A frame-level radar summary also interacts with the IMU representation to form a global motion context that is added back to the radar tokens. The IMU stream therefore provides both local feature conditioning and sequence-level motion information before query decoding.

Structured Hand Prediction: The refined hand queries are decoded into 21-joint poses using a structured head. Rather than directly regressing all joint coordinates, the model predicts the wrist position together with the directions and lengths of the 20 hand bones. Joints are reconstructed recursively as

$$\mathbf{p}_j = \mathbf{p}_{\pi(j)} + \ell_j \mathbf{d}_j,$$

where $\pi(j)$ denotes the parent joint, $\mathbf{d}_j$ the predicted bone direction, and ℓ_j its length. A bounded residual is then added to each joint, $\widehat{\mathbf{P}} = \mathbf{P} + \Delta\mathbf{P}$, with the wrist residual fixed to zero to preserve the predicted root position. Each query additionally produces a hand-objectness score and an auxiliary gesture prediction used only during training.

Sequence-Level Matching and Training Objective: Multiple hand queries are matched to ground-truth trajectories over the complete sequence rather than independently per frame. For each query-hand pair, the matching cost combines 3D joint error, wrist localisation error, hand-objectness score, and inter-frame joint-motion differences, averaged over frames in which the hand is visible. When two hands are present, two distinct queries are selected by minimising the joint sequence-level cost, and the resulting assignments are retained across the sequence. Matched queries are supervised using 3D joint and wrist coordinates, relative-joint and bone constraints, and temporal velocity and acceleration losses. A focal objectness loss separates matched and unmatched queries, while the auxiliary wrist-prior branch is supervised using ground-truth wrist locations. Gesture classification and intermediate decoder predictions are also included. The overall objective is

$$\mathcal{L} = \mathcal{L}_{\text{pose}} + \lambda_{\text{obj}}\mathcal{L}_{\text{obj}} + \lambda_{\text{gesture}}\mathcal{L}_{\text{gesture}} + \lambda_{\text{prior}}\mathcal{L}_{\text{wrist-prior}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}},$$

where $\mathcal{L}_{\text{pose}}$ groups the joint-coordinate, wrist, relative-joint, bone, velocity, and acceleration terms. The individual loss weights are specified in the experimental settings.

Architectural Novelty

A conventional state-space model operates on one-dimensional feature sequences and would require pooling or flattening the spatial radar representation for direct application to hand-pose estimation. Our architecture instead preserves spatial radar tokens and extends the OscSpikeSSM backbone to model radar, IMU, and hand-query dynamics while retaining the spatial and instance-level structure required for multi-hand pose estimation.

(i) Multi-level spiking SSM processing: OscSpikeSSM is used at three levels: to model the temporal evolution of each spatial radar token, encode the synchronised IMU sequence, and refine each hand query across time. This separates the temporal modelling of radar evidence, motion cues, and hand-instance representations while retaining a common state-space formulation.

(ii) Radar-specific spatial modelling: The radar encoder preserves a spatial token grid rather than reducing each frame to a single feature vector, allowing temporal modelling at individual radar locations. An auxiliary wrist-prior branch further extracts hand-localisation cues from a higher-resolution feature map and complements the dense radar representation during decoding.

(iii) Motion-aware multimodal conditioning: Rather than directly concatenating IMU and radar features, the IMU sequence is encoded separately by OscSpikeSSM and used to condition the radar representation through feature modulation and global motion context. This incorporates inertial motion information while retaining radar as the primary source of spatial information.

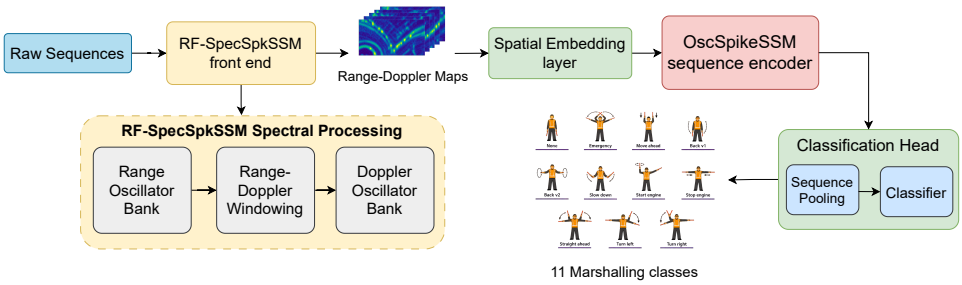

Figure 4: Overview of the RF-SpecSpkSSM radar signal-processing pipeline. Raw radar sequences are processed by the Spike-OSS front end, where range and Doppler oscillator banks replace the corresponding FFT stages to form range-Doppler maps. The maps are spatially embedded and processed by an OscSpikeSSM sequence encoder, followed by sequence pooling and classification into the 11 aircraft-marshalling classes.

(iv) Instance-aware and structured pose decoding: Multiple hands are represented by learnable queries, each forming a hand hypothesis that is refined using its temporal trajectory and radar context. The final 21-joint pose is estimated from the predicted wrist, bone directions, and bone lengths, followed by bounded joint corrections, providing an explicit hand-estimation prior rather than independently regressing all joint coordinates.

Evaluation

Evaluation Protocol: We evaluate on the EgoHands dataset, which provides synchronised mmWave radar, IMU measurements, 3D hand keypoints, and gesture labels. Each recording is uniformly represented using $T = 30$ frames, with up to two hands and 21 joints per hand; a visibility mask handles single-hand sequences. The official split contains 4,151 training and 1,055 test sequences and provides no separate validation set. We therefore use the provided test split for checkpoint selection during training, and report the best performance; no separate held-out test evaluation is performed.

Training Configuration: The model uses feature dimension $D = 256$, an 8×4 radar grid, and $N_q = 16$ hand queries. The frame-level pose decoder contains four layers followed by six global query-context layers with feed-forward dimension 768. Radar and IMU temporal encoders use bidirectional OscSpikeSSM processing with threshold 0.30 and reset amount 0.50. Training is performed for over 700 epochs using AdamW with learning rate 10^{-4}, weight decay 5×10^{-4}, and a 5-epoch warm-up. A batch size of 2 with 16 gradient-accumulation steps gives an effective batch size of 32. OscSpikeSSM parameters use the same learning-rate scale as the rest of the network, query-decoder dropout is 0.15, automatic mixed precision is disabled, and no explicit spike-rate regularisation is applied.

The loss weights are 1.0 for coordinate, wrist, and wrist-relative pose terms; 0.20 for the bone-vector term; 0.10 for bone-length and velocity terms; and 0.02 for acceleration. Objectness, gesture, wrist-prior, and auxiliary decoder losses are weighted by 0.10, 0.20, 0.20, and 0.25, respectively.

Reference Model and Performance Metrics: We compare against the mmEgoHand pose estimator as the task-specific reference. The primary metric is Mean Per Joint Position Error (MPJPE), reported in millimetres. For matched and visible hand trajectories, MPJPE is computed as the mean Euclidean distance between predicted and ground-truth 3D joint coordinates.

A.2.7 Radar Signal Processing

Architecture

Raw ADC Sequence Processing: The model operates directly on raw radar segments $\mathbf{X} \in \mathbb{R}^{B\times 960\times 512}$, where each segment contains 960 chirps with 512 ADC samples per chirp. Fast-time DC removal and a Hann window are first applied independently to every chirp. Rather than comput-

ing a 512-point range FFT, each chirp is passed through a range oscillator bank that directly produces the 41 range coefficients retained by the downstream representation. The resulting complex range features are divided into 17 overlapping slow-time windows of 192 chirps with a hop of 48. After slow-time DC removal and Hann windowing, a second oscillator bank replaces the Doppler FFT, directly producing 31 Doppler coefficients for each retained range bin. The front end, therefore, maps the raw ADC segment to a sequence $\mathbf{X}_{\mathrm{RD}} \in \mathbb{R}^{B\times 17\times 1\times 31\times 41}$ without explicitly computing either the range or Doppler FFT. The selected dimensions correspond to the same physical range and velocity range as those used by the conventional preprocessing pipeline.

Oscillatory Spectral Bank: To replace the fixed Fourier transform while retaining the dynamics of OscSpikeSSM, we derive the spectral front end of RF-SpecSpkSSM from its underdamped oscillatory state-space dynamics. Each stable oscillatory mode can be represented by a discrete complex-conjugate pole, $p = \rho e^{\pm j\phi}$, which gives the recurrent update $\mathbf{z}_n = \rho e^{j\phi}\mathbf{z}_{n-1} + g\mathbf{x}_n$. Here, ρ controls the state decay, ϕ determines the oscillation frequency, and g denotes the input gain. This representation preserves the oscillatory state-space principle of OscSpikeSSM while making each mode directly interpretable as a frequency-selective channel.

For spectral processing, the oscillators are initialised at the required Fourier-bin frequencies, $\phi_m = 2\pi m/N$, following the conventional frequency-bin organisation used in range-Doppler processing (Vermunt & Corradi, 2024). A bank of such oscillators is used to obtain only the range and Doppler coefficients required by the downstream representation. The formulation permits bounded frequency adaptation around these initialised bins; however, in the reported experiments, the oscillator frequencies are kept fixed to preserve the physical range-Doppler grid. The resulting range and Doppler oscillator banks therefore retain the physical organisation of the Fourier spectrum while replacing the fixed transform with recurrent OscSpikeSSM-derived spectral dynamics.

The spiking mechanism of OscSpikeSSM is also retained, but its interaction with the oscillatory state is modified for spectral accumulation. In OscSpikeSSM, reset and adaptation influence the subsequent state trajectory; for spectral estimation, such interruptions would disturb the coherent accumulation of phase and amplitude. RF-SpecSpkSSM therefore removes reset and adaptation from the spectral-state recurrence, allowing the continuous quadrature states to evolve without interruption. Bipolar residual spikes are instead generated from the difference between the continuous oscillator state and a separately reconstructed event state. These positive and negative events update the reconstruction pathway rather than resetting the underlying oscillator.

RF-SpecSpkSSM can therefore be viewed as a spectral specialisation of OscSpikeSSM. It retains the stable oscillatory dynamics and event-based representation of the core model, while modifying the frequency parameterisation, reset behaviour, and adaptation mechanism for range-Doppler analysis. These changes allow the oscillator states to perform coherent frequency-selective accumulation without disrupting the underlying spectral trajectory.

Hybrid Spike-State Spectral Readout: The RF-SpecSpkSSM oscillator bank retains both the continuous state $\mathbf{z}_{\mathrm{state}}$ and its event-decoded approximation $\mathbf{z}_{\mathrm{spike}}$. Instead of fixing the spectral front end to either representation, the output is formed as $\mathbf{z}_{\mathrm{out}} = \alpha\mathbf{z}_{\mathrm{spike}} + (1-\alpha)\mathbf{z}_{\mathrm{state}}$, where α is learned independently for the range and Doppler banks. This allows RF-SpecSpkSSM to determine how strongly each spectral transform should rely on the spike-coded representation. Spike generation is controlled by a single bounded, learnable threshold for the range bank and another for the Doppler bank. The reconstruction quantum is kept fixed, while the thresholds are initially held constant and released later during training. This separates the amount of event activity from the state reconstruction scale, allowing the two spectral stages to learn different firing sensitivities.

Range-Doppler Map Formation: After the RF-SpecSpkSSM Doppler bank, the complex coefficients are converted to magnitude and scaled by the Doppler-window gain. A logarithmic compression is applied, followed by normalisation over the complete 31×41 range-Doppler map. This produces the same spatial representation expected by the downstream classifier, while the frequency coefficients themselves are generated by the RF-SpecSpkSSM oscillator banks rather than FFT operations.

Spiking SSM Gesture Encoder: Each range-Doppler map is first encoded independently by a shared spatial encoder into a compact feature vector. The resulting sequence of spatial embeddings

is then passed through an OscSpikeSSM stack, which models the temporal evolution of the radar signature across the 17 range-Doppler observations. A deterministic leaky readout produces frame-level logits, and the later predictions are pooled to obtain the final sequence-level classification over the 11 aircraft marshalling classes. The spatial encoder preserves the natural 31×41 range-Doppler geometry rather than resizing the maps to a fixed square resolution.

Overall, the architecture applies oscillatory state-space processing at two stages of the radar pipeline. RF-SpecSpkSSM replaces the conventional range and Doppler transforms with frequency-selective oscillator banks, while the downstream OscSpikeSSM stack models the temporal evolution of the resulting range-Doppler sequence. Both components are trained jointly with the final classification objective, yielding an end-to-end radar pipeline in which spectral extraction and temporal modelling are optimised together.

Architectural Novelty

The proposed pipeline extends oscillatory spiking state-space modelling from temporal sequence processing to radar spectral processing. Its main architectural contributions are summarised below:

(i) Spiking state-space spectral transform: Instead of using fixed FFT preprocessing, the under-damped dynamics of the OscSpikeSSM are specialised into frequency-selective oscillator banks. The oscillators are initialised at the required Fourier-bin frequencies and kept fixed in the reported experiments, while the remaining front-end parameters are learned from the task objective. This preserves the physical interpretation of the range and Doppler bins while incorporating the spectral transform into the trainable model.

(ii) Non-disruptive spiking for spectral accumulation: The spiking mechanism is reformulated so that event generation does not perturb the underlying spectral state. This allows the oscillatory SSM to preserve coherent phase and amplitude accumulation while simultaneously maintaining a sparse event representation. Bipolar residual spikes track the continuous oscillator through a separate reconstruction pathway, and the spectral readout can combine the continuous and event-reconstructed states. Thus, spiking is incorporated into the spectral transform without interrupting the dynamics required for frequency-selective accumulation.

(iii) Task-selective spectral computation: Rather than computing the full spectrum and subsequently retaining selected bins, the oscillator banks directly represent only the range and Doppler coefficients required by the downstream task. This yields a task-selective state-space spectral representation while preserving the physical range-Doppler organisation.

(iv) Unified state-space processing from signal to sequence: The same oscillatory state-space principle is used at two stages of the pipeline: RF-SpecSpkSSM forms the range-Doppler representation from raw radar measurements, while the downstream OscSpikeSSM models its temporal evolution. Spectral feature formation and temporal sequence modelling are therefore integrated within a common state-space framework.

Evaluation

Evaluation Protocol: We evaluate the proposed signal-processing architecture on Version 2 of the Aircraft Marshalling Signals dataset (Müller et al., 2023) using only the radar modality. The task contains 11 classes: ten aircraft-marshalling gestures and an additional *none* class. Each sample contains 960 radar chirps with 512 ADC samples per chirp and is divided into 17 overlapping windows of 192 chirps with a hop of 48. The retained range-Doppler region contains 41 range bins spanning approximately 0–6 m and 31 Doppler bins spanning -7.5 to 7.5 m/s. The split contains 13,320 training and 962 test samples. Since no separate validation split is used, the test split is employed for checkpoint selection, with the checkpoint achieving the highest sequence-level accuracy used for reporting; no separate held-out test evaluation is performed.

Training Configuration: The spectral front end uses 41 range and 31 Doppler oscillators with pole radius initialised to $\rho = 0.999$. Their frequencies are initialised at the corresponding Fourier-bin frequencies and kept fixed during the reported experiments to preserve the physical range-Doppler grid. The hybrid continuous-spike readout uses a mixing coefficient initialised to 0.5, and the front-end spike threshold is initialised to 0.05.

Each range-Doppler map is projected to a 128-dimensional representation and processed by two OscSpikeSSM layers with 512 hidden units each. The neuronal threshold and reset amount are both 0.25. Training is performed for 50 epochs using AdamW with learning rates of 3×10^{-4} for the classifier and 2×10^{-5} for the spectral front end, weight decay 5×10^{-4}, and batch size 4. Balanced class weights and label smoothing of 0.05 are used. The objective combines sequence-level classification loss with an auxiliary late-sequence classification loss weighted by 0.10, with no explicit spike-rate regularisation.

Reference Configurations and Performance Metrics: Following the original Aircraft Marshalling benchmark, we use the conventional FFT preprocessing as one of the reference configurations. Raw ADC segments are windowed and transformed using a chirp-wise range FFT followed by a Doppler FFT across chirps, after which the relevant range-Doppler region is retained for classification. We additionally compare against ResNet18, EfficientNet-B1, LeViT-128s, LeViT-128, and MogaNet Tiny as radar-classification baselines, together with the EfficientNet-B1 result reported by Vermunt & Corradi (2024). FFT-SpikingSSM provides the direct controlled comparison with RF-SpecSpkSSM, while the remaining models serve as external classification baselines. Sequence-level classification accuracy is used as the primary performance metric, with the final prediction obtained by averaging the logits over the last three temporal steps.