SNN-FrFT: Adaptive Fractional Representations for Efficient Spiking Forecasting

Anonymous authors
Paper under double-blind review

Abstract

A fixed orthogonal Fourier basis spreads chirp-like signals across coefficients, limiting sparse spiking forecasting. We introduce a spiking neural network with an adaptive fractional Fourier transform (SNN-FrFT). A spiking selector estimates the order from the observed history; the forecasting head predicts residual dynamics, and the same order reconstructs the future phase. Signed Dual-Region Log-Temporal Coding (DR-LTC) covers strong coefficients logarithmically and maps sufficiently weak coefficients to zero-event states. The compact head projects once and reuses current across leaky integrate-and-fire steps. On a three-seed repartitioned NeLoRa benchmark, SNN-FrFT achieves 79.28% spectral peak agreement at spreading factor 10 versus 73.67% for Spiking Fourier Network (SpikF), with 97.23% fewer core multiply-accumulate operations. Synthetic factorization connects the prediction gain to pairing the adaptive input representation with future reconstruction. A matched three-epoch coding study improves NeLoRa peak agreement over linear time-to-first-spike coding while leaving 74.06% of terminal rails silent. On the same HM100, the compiled width-256 cached head uses 38.3% less mean board-level dynamic energy than SpikF's decoder.

1 Introduction

Spiking neural networks (SNNs) encode information through discrete spikes and evolving membrane states. On event-driven neuromorphic hardware, sparse firing reduces spike communication and synaptic accumulation, allowing computation to follow activity (Roy et al., 2019). These properties have motivated SNN applications in temporal prediction, radar processing, and signal detection (Lv et al., 2024b; López-Randulfe et al., 2021; Dakic et al., 2024). Their energy advantage depends on how effectively the input representation exposes sparsity.

Radar, satellite receivers, and battery-powered Internet of Things (IoT) nodes face this requirement directly. Echo arrivals and communication bursts provide reception events around which processing can be triggered, matching the activity-dependent operation of SNNs (López-Randulfe et al., 2021; Dakic et al., 2024). Within each reception window, however, high-rate radio-frequency (RF) samples still require efficient processing. Chirps are common in radar and LoRa chirp spread spectrum (CSS) communication; we use measured LoRa baseband from NeLoRa as our forecasting benchmark (Du et al., 2023).

For chirps, time-domain predictors must learn rapidly changing phase, making frequency-aware representations useful. The fast Fourier transform (FFT) spreads a frequency sweep across coefficients, leaving many channels active. SpikF selects Fourier frequencies but retains this dispersed representation (Wu et al., 2025). The fractional Fourier transform (FrFT) instead adjusts its order to compensate for the sweep and concentrate energy (Almeida, 1994). Across 256 NeLoRa histories, adaptive FrFT reduces the mean support carrying 90% of energy, k_{90}, from 29.46 to 3.83. Concentration does not remove implementation costs: dense discrete FrFT implementations, including trainable fractional layers, require $O(N^2)$ arithmetic and storage (Candan et al., 2000; Koç et al., 2024). The fast algorithm of Ozaktas et al. (1996) achieves $O(N \log N)$ through interpolation, chirp multiplication, and convolution. These operations require arithmetic and intermediate storage regardless of output sparsity. Low-power spiking forecasting therefore needs both efficient transform computation and encoding that converts concentration into fewer events.

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.

We propose **SNN-FrFT**, an adaptive fractional representation and spiking forecasting architecture, to address two challenges. **The first challenge is accurate, sparse coding.** Rate coding spends multiple events on amplitude, while linear time-to-first-spike (TTFS) coding loses near-zero detail when a finite timing budget covers strong peaks (López-Randulfe et al., 2022; Wang et al., 2026). The SNN-based Fourier transform (SNN-FT) improves precision through logarithmically polarized TTFS (LP-TTFS), but assigning later spike times does not itself silence weak nonzero coefficients (Wang et al., 2026). We combine a single-FFT scaled-grid transform with $O(N \log N)$ butterfly connectivity and Signed Dual-Region Log-Temporal Coding (DR-LTC). A coarse logarithmic region covers strong coefficients, while a fine region preserves near-zero detail. A separate terminal gate suppresses weak tails. **The second challenge is adapting the order without future information.** A fixed or globally learned order cannot follow changes between windows. Our Adaptive SNN Alpha Selector (A-SAS) estimates a continuous order from each observed history using changes in instantaneous frequency. That same order controls input adaptation and future-phase reconstruction, so the predictor learns residual dynamics after the estimated sweep is removed.

Our contributions are:

- **Adaptive representation with matched reconstruction.** We couple history-based order selection to future-phase reconstruction. Their interaction accounts for 98.92% of the improvement in a controlled chirp ablation.
- **Sparse fractional computation and coding.** The structured transform and DR-LTC connect energy concentration to event sparsity, leaving 74.06% of terminal scalar channels silent in the matched coding study.
- **Efficient forecasting with hardware evidence.** The head projects once and reuses current over 16 leaky integrate-and-fire (LIF) steps. At spreading factor (SF) 10, it improves peak agreement by 5.61 percentage points over SpikF with 97.23% fewer core multiply-accumulate operations (MACs). On HM100 at batch 32, the compiled width-256 cached head uses 38.3% less mean board-level dynamic energy and delivers $2.07\times$ the throughput of SpikF's decoder.

2 Related Work

2.1 Spiking prediction and input representations

Spiking forecasters adapt convolutional networks, recurrent networks, and Transformers through spike encoders and temporal alignment (Lv et al., 2024b). Spikformer introduces spiking attention (Zhou et al., 2023), and central pattern generators add temporal structure (Lv et al., 2024a). These operators retain the input representation when the signal has changing frequency.

Spiking Wavelet Transformer uses multiscale representations (Fang et al., 2024); SpikF selects frequencies within a fixed Fourier basis (Wu et al., 2025). DLinear (Zeng et al., 2023) and the Frequency Interpolation Time Series Analysis Baseline (FITS) (Xu et al., 2024) establish compact conventional heads. We couple an adaptive chirp representation with future reconstruction and measure its prediction benefit separately from head cost.

2.2 Fast and learnable fractional Fourier transforms

The FrFT rotates time–frequency distributions (Almeida, 1994). Fast sampled algorithms (Ozaktas et al., 1996), butterfly factorization (Cooley & Tukey, 1965; Dao et al., 2019), and discrete definitions (Candan et al., 2000) provide computational and algebraic foundations. Trainable layers and DeepFRFT optimize fractional representations (Koç et al., 2024; Yu et al., 2023); our frozen selector supplies an input-dependent order for each window.

Our pre- and post-chirps surround a fixed radix-2 FFT on an order-dependent grid. Building on structured spiking Fourier computation (López-Randulfe et al., 2021; 2022), we pair the input representation with future-phase reconstruction.

2.3 Temporal coding, signed values, and dynamic range

Temporal coding places magnitude in spike latency (Mostafa, 2018; López-Randulfe et al., 2022). SNN-FT combines signed logarithmically polarized time-to-first-spike (LP-TTFS) coding, piecewise ternary spiking neurons (PTSNs), and radix-4 connectivity (Wang et al., 2026). Differential coding transfers numerical information between artificial neural network (ANN) and SNN computation (Huang et al., 2025); quantization methods allocate precision across value ranges (Gray & Neuhoff, 1998).

DR-LTC allocates resolution through coarse logarithmic and shifted fine regions plus a terminal magnitude gate. Section 4.5 compares the LP-TTFS response under a common forecasting head, isolating coding-level behavior from SNN-FT's complete architecture.

3 Methodology

3.1 Adaptive input representation

We forecast $H = 256$ complex in-phase/quadrature (I/Q) samples from a history $x \in \mathbb{C}^N$ with $N = 1024$, normalized by its root mean square (RMS). A-SAS reads the complete history to select one order $\alpha \in (0.25, 1.75)$, which the transform and output restoration share throughout the window (Figure 1). Each new window resets the selector and prediction-head membrane states and receives a fresh order estimate.

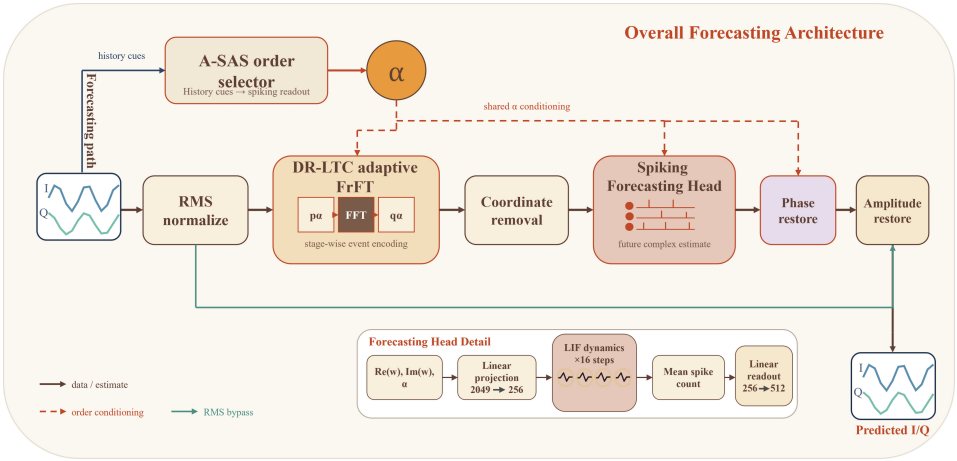

Figure 1: SNN-FrFT with a shared history-conditioned order and a current-reuse forecasting head.

Let $\phi = \pi\alpha/2$. The FrFT adapts the representation to a chirp's changing frequency (Almeida, 1994). A chirp $x_n = A\exp[\mathrm{i}(2\pi f_0 n + \pi c n^2/N + \varphi)]$ loses its quadratic phase after multiplication by $\exp(\mathrm{i}\pi n^2 \cot\phi/N)$ when $c + \cot\phi = 0$. This relation guides order selection: remove the estimated quadratic phase from the history, then restore the phase specified by the same order at future time indices.

We normalize the history by its RMS,

$$r = \sqrt{\frac{1}{N}\sum_{n=-N/2}^{N/2-1} \|x_n\|^2}. \tag{1}$$

For centered indices $n, k = -N/2, \ldots, N/2-1$, our sampled transform is

$$z_k = d_\alpha(k)\left[\mathcal{F}_N\left\{(x_n/r)e^{\mathrm{i}\pi n^2\cot\phi/N}\right\}\right]_k, \tag{2}$$

$$d_\alpha(k) = \frac{e^{-\mathrm{i}(\pi/4-\phi/2)}}{\sqrt{\sin\phi}} e^{\mathrm{i}\pi k^2 \sin\phi\cos\phi/N}.$$

Here $\mathcal{F}_N$ is the centered orthonormal discrete Fourier transform (DFT). Sampling the FrFT kernel at $t_n = n/\sqrt{N}$ and $u_k = k \sin\phi/\sqrt{N}$ gives $u_k t_n \csc\phi = kn/N$ and $u_k^2 \cot\phi = k^2 \sin\phi \cos\phi/N$: a pre-chirp, centered FFT, and post-chirp on an order-dependent grid. Before quantization, $\alpha = 1$ recovers the DFT. Radix-2 factorization (Cooley & Tukey, 1965) uses $N/2$ butterflies per stage over $\log_2 N$ stages; the chirp diagonals add $O(N)$ connections.

3.2 Independent Spiking Order Selection

A-SAS extracts ten history features: log RMS, log peak-to-RMS ratio, and phase-curvature estimates with their coherence at lags $1, 2, 4, 8$. A $10 \to 32 \to 64$ projection with tanh and sigmoid activations drives 64 integrate-and-fire units for 32 ticks. The readout combines normalized spike counts $\mathbf{c}/32$ and residual membrane potentials $\mathbf{v}_{32}/32$:

$$\alpha = 0.25 + 1.5\,\sigma\big(\log 0.2 + \mathbf{w}^\top[\mathbf{c}/32, \mathbf{v}_{32}/32] + b\big)\,. \tag{3}$$

The residual supplies continuous information beyond count resolution. Zero-initialized readout weights and bias give $\alpha = 0.5$ initially. The selector has 2,593 parameters and executes once per window.

Its supervision comes from phase curvature in the observed history. With $p_n = x_n/|x_n|$, define C_ℓ as the mean of $p_{n+2\ell} p_n (p^*_{n+\ell})^2$ over valid indices. We optimize

$$\mathcal{L}_{\text{focus}} = \frac{1}{2} \sum_{\ell \in \{8,16\}} \frac{1 - \mathrm{Re}[C_\ell e^{\mathrm{i} s_\ell \cot(\pi\alpha/2)}]}{s_\ell^2}, \qquad s_\ell = 2\pi\ell^2/N. \tag{4}$$

For an ideal chirp, $C_\ell = e^{\mathrm{i} s_\ell c}$, so the loss rewards cancellation of quadratic phase. It uses neither future samples nor generating-order labels. We learn this mapping on synthetic histories and adapt it to NeLoRa training histories, then freeze its parameters during head training. The order remains input dependent after freezing.

3.3 Dual-Region Log-Temporal Coding

DR-LTC extends signed logarithmic timing (Wang et al., 2026): each real or imaginary rail carries a sign, an event time, and an activity mask (Figure 2). With 64 ticks and a boundary at tick 32, the response is

$$R(\tau) = \begin{cases} 2^{5(1-\tau/32)}, & 0 \le \tau \le 32, \\ 2^{(64-\tau)/32} - 1, & 32 < \tau \le 64. \end{cases} \tag{5}$$

The first region spans magnitudes from 32 down to one, and the second reaches zero. For $a = |v|/U$ clipped to 32, the inverse map is $\tau = 32(5 - \log_2 a)/5$ for $a \ge 1$, and $\tau = 64 - 32\log_2(1 + a)$ otherwise. Integer rounding gives $\widehat{v} = \mathrm{sign}(v) U R(\mathrm{round}\,\tau)$. Values reconstructed as zero emit no event. A terminal gate additionally removes coefficients with complex magnitude at most $0.03U$.

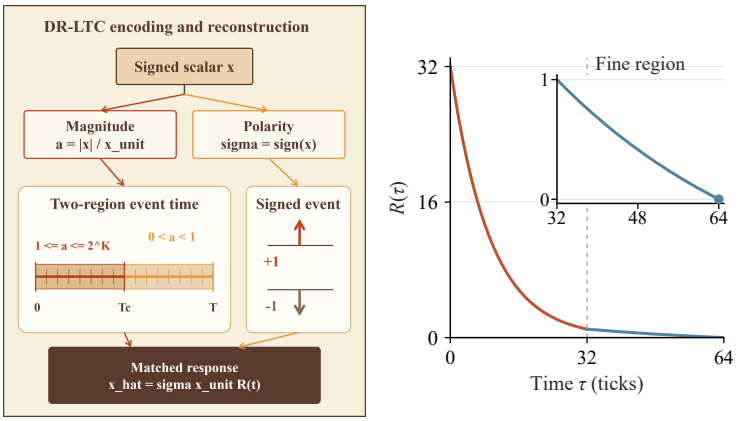

Figure 2: DR-LTC encoding flow (left) and response with a fine-region inset (right).

The responses meet at $R(32) = 1$, with endpoints $R(0) = 32$ and $R(64) = 0$. For unclipped values rounded within one region, the half-tick timing error bounds relative error by $2^{5/64} - 1$ in the coarse region and absolute error by $2U(2^{1/64} - 1)$ in the fine region. With 64 ticks over $[0, 32U]$, linear TTFS rounds magnitudes below $U/4$ to zero. The DR-LTC fine region instead reaches this cutoff at $U(2^{1/64} - 1) \approx 0.01089U$, before the separate complex-magnitude gate is applied. This finer near-zero resolution can retain coefficients discarded by linear timing.

Each diagonal or butterfly stage applies DR-LTC coding and reconstructs its inputs before weighted propagation. The stage unit U is fixed from the maximum training-calibration rail magnitude divided by eight. Signed rails preserve interference between complex components, while the activity mask makes a zero-event coefficient explicit. The software backend applies structured arithmetic to reconstructed values and records emitted events separately from matrix operations.

3.4 Compact head and future reconstruction

After terminal reconstruction, we divide coefficients by the known factor $d_\alpha(k)$. Concatenating real and imaginary components with α produces a 2049-dimensional head input. A $2049 \to 256$ linear layer computes a current j once. Sixteen LIF steps reuse that current:

$$v_t^- = 0.9v_{t-1} + j, \quad s_t = \mathbf{1}[v_t^- \geq 0.5], \quad v_t = v_t^- - 0.5s_t, \quad b = W_2 \left(\tfrac{1}{16} \sum_{t=1}^{16} s_t \right) + b_2. \tag{6}$$

The $256 \to 512$ readout predicts 256 complex residual samples. For future centered indices $q = 512, \ldots, 767$, we reconstruct the signal using the same order selected for the input:

$$\widehat{y}_q = r\, b_q \exp[-\mathrm{i}\pi q^2 \cot(\pi\alpha/2)/N]. \tag{7}$$

The head learns the future residual sequence; reconstruction restores the quadratic phase removed from the history, evaluated at future indices, and the history RMS scale. Mean absolute error (MAE) averages componentwise real and imaginary errors. Surrogate gradients (Neftci et al., 2019; Zenke & Ganguli, 2018) train the head with $\mathcal{L}_{\text{head}} = \mathrm{MAE}(\widehat{y}, y) + 0.05\,\overline{r}\,\overline{s}$, where $\overline{r}$ is the batch-mean history RMS, and $\overline{s}$ is the mean hidden firing rate.

Event and matrix accounting. At coding interface ℓ, each scalar rail emits at most one event. For B windows, C_ℓ rails, activity m_{bi}^ℓ, and outgoing nonzero synaptic fan-out f_{bi}^ℓ, define

$$E_\ell = \frac{1}{B} \sum_{b,i} m_{bi}^\ell, \qquad Z_\ell = 1 - \frac{E_\ell}{C_\ell}, \qquad S_\ell = \frac{1}{B} \sum_{b,i} m_{bi}^\ell f_{bi}^\ell. \tag{8}$$

Here Z_ℓ is the silent fraction and S_ℓ counts synaptic deliveries under direct event routing; uniform fan-out w gives $S_\ell = wE_\ell$. Terminal division by $d_\alpha(k)$ mixes real and imaginary rails, preserving zero complex coefficients but not individual rail masks. Our backend reconstructs values and executes dense projections, whose matrix count is independent of E_ℓ. Current reuse requires one input projection, $16w$ neuron updates, and one count-readout matrix product. Figure 6 measures coding events separately from MACs and energy.

For real multivariate series, channels share a head. History-only instance normalization and 16 zeros on each side convert 96 observations into a 128-point real input. A history Hilbert representation supplies selector cues, while the data path uses the real observations. The $257 \to 256 \to 128$ head predicts normalized real values, crops 96 outputs, and reverses instance normalization. This real-valued path does not use the I/Q-specific future chirp restoration.

3.5 Sensitivity and computational structure

For $\psi_\alpha(q) = -\pi q^2 \cot(\pi\alpha/2)/N$, the restoration sensitivity is

$$\frac{\partial \psi_\alpha(q)}{\partial \alpha} = \frac{\pi^2 q^2}{2N} \csc^2(\pi\alpha/2). \tag{9}$$

At $\alpha = 1$, an order perturbation of 10^{-3} changes phase by approximately 1.26–2.84 radians across the I/Q horizon to first order. For a fixed exact residual of amplitude A_q, phase error $\Delta\psi_q$ contributes

$A_q^2(1 - \cos \Delta\psi_q)$ to mean squared real/imaginary error. Freezing the history-trained selector keeps the mapping from each history to its input representation and future reconstruction fixed as the head learns.

Executing each spatial projection once gives an I/Q core matrix cost of $(2N + 1)w + 2Hw$ MACs, plus 2,496 selector MACs. At $w = 256$, this is 658,112 MACs with 658,977 total parameters, including the frozen selector. For real data, the corresponding total is 101,056 MACs per variable-window. Including the fixed transform's $O(N \log N)$ operations and the membrane's $O(Tw)$ updates gives total I/Q arithmetic $O(N \log N + (N + H)w + Tw)$, apart from the small selector.

4 Experiments

4.1 Experimental Setup

Datasets and baselines. We evaluate NeLoRa SF8–10 baseband sampled at 1 MHz (Du et al., 2023) and Synthetic Chirp ($N = 1024$, $H = 256$), plus the Electricity Transformer Temperature (ETT) datasets ETTh1 and ETTm1, and Weather ($L = H = 96$). NeLoRa retains training packets and repartitions prior development packets into disjoint validation/test sets, with a shared frozen selector. We compare width-256 heads with SpikF (Wu et al., 2025) at 16 simulation steps.

Metrics. For the raw, centered 256-point FFT peak $p(y) = \arg\max_k |\,\mathrm{FFT}(y)_k|^2$, we report peak agreement $\mathbb{E}[\mathbf{1}\{p(\widehat{y}) = p(y)\}]$ and noncircular peak-bin error $\mathbb{E}[|p(\widehat{y}) - p(y)|]$, before dechirping or symbol decoding. I/Q MAE and mean squared error (MSE) average componentwise errors; macroscopic MAE uses training-StandardScaler coordinates. Concentration k_{90} is the smallest energy-ranked support containing 90% of energy.

Compute. Core MACs count linear/convolutional execution, including A-SAS and temporal expansion. Fixed spectral arithmetic, coding, normalization, and membrane updates are separate costs. Graphics processing unit (GPU) timing in Figure 5 covers the frontend and head with device-resident inputs; Section 4.6 measures cached heads on both devices with transfers and host restoration.

4.2 Real-World RF Baseband: NeLoRa Three-Seed Benchmark

Table 1: NeLoRa spectral localization and core MACs (mean ± sample standard deviation (SD), three seeds). CFO averages six offsets on SF10; peak gains are percentage points (pp).

Condition	Metric	SpikF	SNN-FrFT	Change vs. SpikF
All SFs	Core MACs/window	23,756,800	**658,112**	−97.23%
SF8 (clean)	Peak (%) ↑	$\mathbf{100.00 \pm 0.00}$	98.41 ± 0.80	Baseline at ceiling
	Error (bins) ↓	$\mathbf{0.0000 \pm 0.0000}$	0.0239 ± 0.0069	+0.0239 bins
SF9 (clean)	Peak (%) ↑	$\mathbf{100.00 \pm 0.00}$	99.92 ± 0.13	Baseline at ceiling
	Error (bins) ↓	$\mathbf{0.0000 \pm 0.0000}$	0.0008 ± 0.0013	+0.0008 bins
SF10 (clean)	Peak (%) ↑	73.67 ± 0.46	$\mathbf{79.28 \pm 0.79}$	+5.61 pp
	Error (bins) ↓	0.2633 ± 0.0046	$\mathbf{0.2072 \pm 0.0079}$	−21.29% error
SF10 (+CFO)	Peak (%) ↑	72.21 ± 0.61	$\mathbf{76.06 \pm 0.16}$	+3.85 pp
	Error (bins) ↓	$\mathbf{0.5299 \pm 0.0678}$	0.6180 ± 0.0354	+16.62% error

On SF10, SNN-FrFT improves peak agreement by 5.61 percentage points over SpikF and leads in all three paired runs (Table 1). Across SF8–10, agreement rises from 91.36% to 92.78%, and mean peak-bin error falls from 0.0864 to 0.0742 bins.

SpikF reaches 100% peak agreement on clean SF8 and SF9, while SNN-FrFT reaches 98.41% and 99.92%; its localization advantage is concentrated on SF10. For the CFO test, we apply a continuous phase ramp to each 1280-sample history–future pair at offsets ±1953.125, ±3906.25, and ±7812.5 Hz, corresponding to ±0.5, ±1, and ±2 future-FFT bins. Models and coding scales remain frozen. Averaging the six offsets equally, SNN-FrFT improves peak agreement from 72.21% to 76.06%, while mean peak-bin error increases from 0.5299 to 0.6180. These controlled perturbations of measured I/Q support improved exact peak agreement, not uniformly smaller localization errors.

Figure 3 shows the measured accuracy–cost tradeoff; open markers denote individual seeds and filled markers denote their means. SNN-FrFT executes 658,112 core matrix MACs per window against SpikF's 23,756,800, reducing this cost by 97.23%. Fixed FFT uses 655,616 MACs and reaches 75.48% SF10 agreement; adaptive input representation with matching future reconstruction raises agreement to 79.28% with 2496 additional selector MACs. The width-256 model has 658,977 parameters versus SpikF's 471,613. Computing each head projection once, then reusing its current across the LIF dynamics, accounts for the lower execution count.

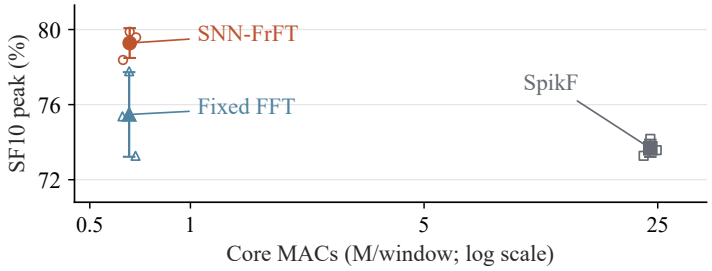

Figure 3: NeLoRa accuracy–cost tradeoff: three seeds and mean ± sample SD.

4.3 Input Representation and Future Reconstruction: Chirp Ablation

We vary the adaptive input representation and future-phase restoration independently, sharing head width, one paired initialization, data, and stopping rules across four configurations and 1024 test windows. Fixed FFT without restoration gives MAE 0.645774. Input adaptation alone yields 0.642669; restoration alone yields 0.647458. Combining both gives 0.514547, a 20.32% reduction, while input concentration improves from $k_{90} = 257.397$ to 12.266 (Figure 4).

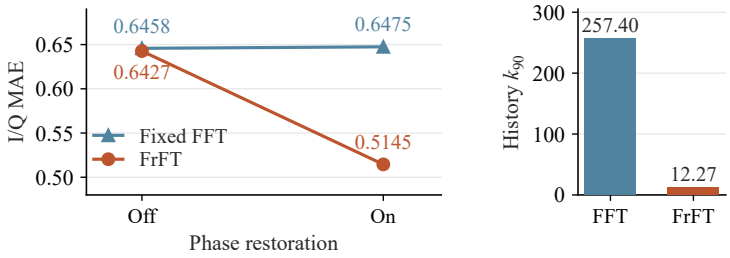

Figure 4: Chirp forecasting with input adaptation and future-phase restoration, separately and together, with history concentration shown alongside.

With superscript R denoting restoration, the interaction is

$$I = E_{\mathrm{FrFT}} + E^{\mathrm{R}}_{\mathrm{FFT}} - E^{\mathrm{R}}_{\mathrm{FrFT}} - E_{\mathrm{FFT}} = 0.129807. \tag{10}$$

It accounts for $I/(E_{\mathrm{FFT}} - E^{\mathrm{R}}_{\mathrm{FrFT}}) = 98.92\%$ of the total reduction. In this experiment, the forecasting benefit depends on pairing the adaptive input representation with matching future reconstruction. FrFT-input configurations also pass the selected order to the head; FFT-input configurations fix that feature at one and obtain any restoration order separately from A-SAS. This intervention tests the complete input-representation path and its coupling to future reconstruction.

4.4 Cross-Domain Evaluation: Head Efficiency and the Fourier Limit

ETTh1, ETTm1, and Weather test whether the forecasting head remains efficient beyond chirp signals and how A-SAS behaves without an imposed frequency sweep. We use $L = H = 96$ and evaluate both prediction error and execution cost (Figure 5).

Head efficiency across domains. SNN-FrFT uses 101,056 core MACs per variable window versus SpikF's 25,709,568 on ETT (99.61% fewer). The fixed-FFT compact head achieves ETTh1

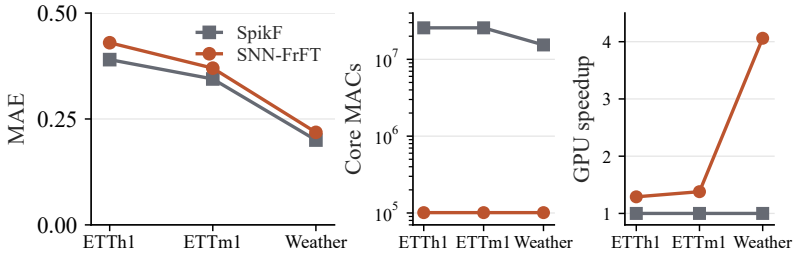

Figure 5: Cross-domain comparison on ETTh1, ETTm1, and Weather. The middle and right panels report core MACs and GPU speedup.

MAE 0.389668 with 98,560 MACs, versus SpikF's 0.390121, isolating head efficiency from fractional adaptation. Complete frontend-and-head GPU speedups are $1.29\times$, $1.38\times$, and $4.06\times$ on ETTh1, ETTm1, and Weather at batch 32.

Order selection near the Fourier limit. For an ideal chirp with zero sweep rate, $c+\cot(\pi\alpha/2) = 0$ gives $\alpha = 1$, the Fourier basis. A-SAS selects mean orders of 0.9850, 0.9839, and 0.9794 on the three datasets, respectively; 62.13%, 80.70%, and 91.31% of windows satisfy $|\alpha - 1| \leq 0.1$. The history-based selector therefore favors orders near the Fourier limit, with the strongest concentration on Weather and wider order variation on ETTh1.

SNN-FrFT yields MAE 0.429986, 0.370008, and 0.218420, versus SpikF's 0.390121, 0.344471, and 0.200232. These 10.22%, 7.41%, and 9.08% increases separate the head's computational advantage from the chirp-specific representation gains.

4.5 DR-LTC Ablation

Table 2: Matched DR-LTC coding ablation. Mean ± sample SD over three seeds; all models use the same selector, width-256 head, train-only calibration, and three training epochs.

Dataset	Coding	MAE ↓	MSE ↓	Peak (%) ↑	SF10 peak (%) ↑	Error (bins) ↓
Chirp	Linear TTFS	0.630808 ± 0.000691	0.547503 ± 0.005899	7.94 ± 0.91	n/a	6.7155 ± 0.3394
	DR-LTC	**0.548753 ± 0.000625**	**0.437899 ± 0.001201**	11.65 ± 0.85	n/a	**6.5566 ± 0.0282**
NeLoRa	Linear TTFS	0.010343 ± 0.000035	0.000192 ± 0.000001	78.65 ± 0.49	70.47 ± 0.46	0.3195 ± 0.0302
	DR-LTC	**0.006378 ± 0.000042**	**0.000091 ± 0.000001**	**90.87 ± 0.60**	**74.47 ± 1.67**	**0.0992 ± 0.0060**

Coding comparisons share 64 ticks, real/imaginary sign rails, training-derived scales, width-256 heads, paired seeds, and three training epochs (Table 2). Only full DR-LTC uses the $0.03U$ terminal gate; its comparison with linear and single-region log coding changes both response and gating. The DR-LTC gate-off control isolates the terminal threshold (Figure 6).

DR-LTC improves chirp MAE and NeLoRa localization over linear TTFS (Table 2). On 256 training histories, its terminal weak-coefficient reconstruction MSE falls from 0.003400 to 0.000220, consistent with its finer near-zero response.

Where events become sparse. Frontend event counts are identical across the three head seeds. DR-LTC emits 531.2 terminal events per window, leaving 74.06% of 2048 scalar rails silent (Figure 6). Across 13 coding interfaces, it emits 21,349.4 events out of 26,624 rail opportunities. Silence grows from 0.10% before the first butterfly to 48.97% before the post-chirp and 74.06% at the terminal interface, locating sparsity near the transform output.

The single-region log control uses the signed LP-TTFS response (Wang et al., 2026) without terminal gating. DR-LTC emits **48.11% fewer events** (531.2 versus 1023.8); against its gate-off control (1111.4), the reduction is 52.20%. SF10 agreement is 74.47% versus 75.18% and 75.08% for these logarithmic controls, with differences within one reported SD. Linear TTFS emits fewer events (331.6), but its coarser near-zero truncation accompanies lower SF10 agreement (70.47%) and greater bin error (0.3195 versus 0.0992). This response-level comparison does not implement SNN-FT's PTSN dynamics or radix-4 connectivity.

Terminal counts precede phase removal; synaptic operations also depend on real/imaginary mixing and receiving connectivity (Equation (8)).

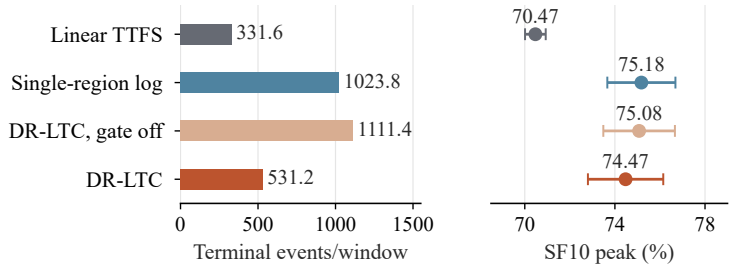

Figure 6: Terminal events and SF10 localization; error bars show sample SD over three head seeds.

4.6 Neuromorphic Silicon Deployment: Lynxi HM100 Profiling

We profile SpikF and SNN-FrFT cached heads on GTX 1650 and HM100 at batch 32 over 2048 historical NeLoRa windows, using single-seed checkpoints distinct from the three-seed benchmark. Both frontends are excluded; transfers and original host restoration are included. Table 3 reports board-level energy and host-inclusive throughput.

Table 3: Cached heads at batch 32, mean ± SD over three interleaved measurement repeats per device. Energy ratios are SpikF/head mean dynamic energy on the same device.

Head	Dynamic (mJ/window)	Total (mJ/window)	Throughput (windows/s)	Energy ratio
GTX 1650				
SpikF	2.5770 ± 0.0281	3.2915 ± 0.0418	17152.2 ± 305.5	$1.00\times$
SNN-FrFT (128)	0.5343 ± 0.0384	1.2405 ± 0.0239	17332.7 ± 367.9	$4.82\times$
SNN-FrFT (256)	0.6387 ± 0.0214	1.3339 ± 0.0469	17589.7 ± 668.4	$4.03\times$
HM100				
SpikF	0.1140 ± 0.0241	2.2542 ± 0.0305	4390.7 ± 39.3	$1.00\times$
SNN-FrFT (128)	0.0726 ± 0.0031	1.1048 ± 0.0066	9134.7 ± 44.7	$1.57\times$
SNN-FrFT (256)	0.0703 ± 0.0102	1.1020 ± 0.0346	9088.2 ± 308.6	$1.62\times$

Same-board HM100 comparison. The width-256 head reduces mean dynamic energy from 0.1140 to 0.0703 mJ/window, a 38.3% reduction ($1.62\times$ energy ratio), and increases sustained throughput from 4390.7 to 9088.2 windows/s ($2.07\times$). Width 128 gives comparable results. SpikF's decoder processes all 16 cached feature steps, whereas our head reuses one projected current. The packed inputs also differ: 32,772 versus 2050 float32 values/window. The measured comparison includes these native transfer costs.

GPU and cross-device comparison. On GTX 1650, width 256 uses 75.2% less mean dynamic energy than SpikF, with comparable sustained throughput. GPU/HM100 dynamic-energy ratios are $7.36\times$ and $9.08\times$ for the same width-128/256 heads. All three HM100 heads fail numerical-equivalence checks against their software counterparts, so the results characterize compiled deployments.

5 Conclusion and Future Work

SNN-FrFT combines a history-adaptive representation with same-order future reconstruction, sparse DR-LTC encoding, and current reuse. It improves NeLoRa SF10 peak agreement by 5.61 percentage points with 97.23% fewer core matrix MACs than SpikF. Its compiled width-256 cached head reduces mean board-level dynamic energy by 38.3% against SpikF's decoder on HM100.

Future work targets multi-component frequency-modulated continuous-wave (FMCW) radar and acoustic Doppler tracking with multi-order A-SAS. Deploying the frontend, head, and reconstruction on one neuromorphic device would enable full-pipeline energy measurement with fewer host–device transfers.

AI USE STATEMENT

During the preparation of this manuscript, we used AI tools' assistance for manuscript formatting, grammar, LaTeX table styling, and prose polishing. All theoretical derivations, algorithmic designs, experimental evaluations, hardware measurements, and final claims were conceived, executed, verified, and approved solely by us.

REPRODUCIBILITY STATEMENT

The manuscript specifies the transform, order selector, coding rule, prediction head, and reconstruction through their equations and execution order. Experimental configurations include dataset partitions, preprocessing, random seeds, training budgets, and checkpoint-selection rules. Hardware protocols specify the cached-head boundary, warm-up, power sampling, idle-power subtraction, and numerical checks. Reported standard deviations distinguish variation across forecasting-head seeds from variation across repeated hardware measurements.

REFERENCES

L. B. Almeida. The fractional Fourier transform and time-frequency representations. *IEEE Transactions on Signal Processing*, 42(11):3084–3091, 1994. doi: 10.1109/78.330368.

Cagri Candan, M. Alper Kutay, and Haldun M. Ozaktas. The discrete fractional Fourier transform. *IEEE Transactions on Signal Processing*, 48(5):1329–1337, 2000. doi: 10.1109/78.839980.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. doi: 10.1090/S0025-5718-1965-0178586-1.

Kosta Dakic, Bassel Al Homssi, Sumeet Walia, and Akram Al-Hourani. Spiking neural networks for detecting satellite internet of things signals. *IEEE Transactions on Aerospace and Electronic Systems*, 60(1):1224–1238, 2024. doi: 10.1109/TAES.2023.3334216.

Tri Dao, Albert Gu, Matthew Eichhorn, Atri Rudra, and Christopher Ré. Learning fast algorithms for linear transforms using butterfly factorizations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1517–1527. PMLR, 2019. URL https://proceedings.mlr.press/v97/dao19a.html.

Jialuo Du, Yidong Ren, Mi Zhang, Yunhao Liu, and Zhichao Cao. NELoRa-Bench: A benchmark for neural-enhanced LoRa demodulation. In *ICLR Workshop on Machine Learning for IoT*, 2023. doi: 10.48550/arXiv.2305.01573. URL https://arxiv.org/abs/2305.01573.

Yuetong Fang, Ziqing Wang, Lingfeng Zhang, Jiahang Cao, Honglei Chen, and Renjing Xu. Spiking wavelet transformer. In *Computer Vision – ECCV 2024*, pp. 19–37. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-73116-7_2.

Robert M. Gray and David L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, 1998. doi: 10.1109/18.720541.

Zihan Huang, Wei Fang, Tong Bu, Peng Xue, Zecheng Hao, Wenxuan Liu, Yuanhong Tang, Zhaofei Yu, and Tiejun Huang. Differential coding for training-free ANN-to-SNN conversion. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25235–25264. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25i.html.

Emirhan Koç, Tuna Alikaşifoğlu, Arda Can Aras, and Aykut Koç. Trainable fractional Fourier transform. *IEEE Signal Processing Letters*, 31:751–755, 2024. doi: 10.1109/LSP.2024.3372779.

Javier López-Randulfe, Tobias Duswald, Zhenshan Bing, and Alois Knoll. Spiking neural network for Fourier transform and object detection for automotive radar. *Frontiers in Neurorobotics*, 15:688344, 2021. doi: 10.3389/fnbot.2021.688344.

Javier López-Randulfe, Nico Reeb, Negin Karimi, Chen Liu, Hector A. Gonzalez, Robin Dietrich, Bernhard Vogginger, Christian Mayr, and Alois Knoll. Time-coded spiking Fourier transform in neuromorphic hardware. *IEEE Transactions on Computers*, 71(11):2792–2802, 2022. doi: 10.1109/TC.2022.3162708.

Changze Lv, Dongqi Han, Yansen Wang, Xiaoqing Zheng, Xuanjing Huang, and Dongsheng Li. Advancing spiking neural networks for sequential modeling with central pattern generators. In *Advances in Neural Information Processing Systems*, volume 37, pp. 26915–26940, 2024a. doi: 10.52202/079017-0845. URL https://papers.nips.cc/paper_files/paper/2024/hash/2f55a8b7b1c2c6312eb86557bb9a2bd5-Abstract-Conference.html.

Changze Lv, Yansen Wang, Dongqi Han, Xiaoqing Zheng, Xuanjing Huang, and Dongsheng Li. Efficient and effective time-series forecasting with spiking neural networks, 2024b. URL https://arxiv.org/abs/2402.01533.

Hesham Mostafa. Supervised learning based on temporal coding in spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3227–3235, 2018. doi: 10.1109/TNNLS.2017.2726060.

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.

Haldun M. Ozaktas, Orhan Arikan, M. Alper Kutay, and Gozde Bozdagi. Digital computation of the fractional Fourier transform. *IEEE Transactions on Signal Processing*, 44(9):2141–2150, 1996. doi: 10.1109/78.536672.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019. doi: 10.1038/s41586-019-1677-2.

Shuai Wang, Haorui Zheng, Yukun Chen, Ammar Belatreche, Guoqing Wang, Yeying Jin, Jibin Wu, Malu Zhang, Yang Yang, and Haizhou Li. SNN-FT: Temporal-coded spiking neural networks for Fourier transform. *IEEE Transactions on Neural Networks and Learning Systems*, 37(4):1618–1630, 2026. doi: 10.1109/TNNLS.2025.3622893.

Wenjie Wu, Dexuan Huo, and Hong Chen. SpikF: Spiking Fourier network for efficient long-term prediction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67342–67368. PMLR, 2025. URL https://proceedings.mlr.press/v267/wu25m.html.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with 10k parameters. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.03756.

Hu Yu, Jie Huang, Lingzhi Li, Man Zhou, and Feng Zhao. Deep fractional Fourier transform. In *Advances in Neural Information Processing Systems*, volume 36, pp. 72761–72773, 2023. doi: 10.52202/075280-3181. URL https://papers.nips.cc/paper_files/paper/2023/hash/e66309ead63bc1410d2df261a28f602d-Abstract-Conference.html.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.

Friedemann Zenke and Surya Ganguli. SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, 30(6):1514–1541, 2018. doi: 10.1162/neco_a_01086.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=frE4fUwz_h.

A Experimental Configurations

A.1 Datasets and packet partitions

Both complex forecasting tasks use histories of $N = 1024$ samples and futures of $H = 256$ samples. Synthetic Chirp covers both sweep directions, amplitudes 0.7–1.3, and additive complex noise at 20 dB signal-to-noise ratio (SNR). NeLoRa (Du et al., 2023) uses synchronized, carrier-frequency-offset-compensated baseband sampled at 1 MHz with 125 kHz bandwidth. Table 4 lists window counts.

Table 4: Forecasting window counts. The three macroscopic datasets use $L = H = 96$.

Dataset	Training	Validation	Test	Variables / signal type
Synthetic Chirp	8192	2048	1024	Complex I/Q
NeLoRa	8192	1033	1015	Complex I/Q, SF8–10
ETTh1	8449	2785	2785	7 variables, hourly
ETTm1	34369	11425	11425	7 variables, 15-minute
Weather	36696	5175	10444	21 variables

NeLoRa retains its 8192 training windows and repartitions the former 2048-window development set into validation and test sets. The global packet-label sets are

$$P_{\mathrm{val}} = \{0, 5, 10, 15, 20, 25, 65, 70\},$$
$$P_{\mathrm{te}} = \{30, 35, 40, 45, 50, 55, 60\},$$
$$P_{\mathrm{tr}} = \{0, 1, \ldots, 72\} \setminus (P_{\mathrm{val}} \cup P_{\mathrm{te}}).$$

The sets have 58, eight, and seven labels, respectively, and all three pairwise intersections are empty. They yield 155/20/20 $(\mathrm{SF}, \mathrm{packet\ identifier})$ groups in training/validation/test, with source files also disjoint. The test split contains 251 SF8, 431 SF9, and 333 SF10 windows from six, seven, and seven SF-specific packet groups. This establishes packet isolation for the reported head comparison. Earlier method development used the source development population, so the repartition is not a historically untouched test population. The three head seeds share a frozen A-SAS selector and fixed coding scales; validation MAE selects the forecasting checkpoint.

The SF10 CFO comparison reuses all 333 SF10 test windows. A continuous phase ramp at ± 1953.125, ± 3906.25, or ± 7812.5 Hz is applied to the concatenated 1280-sample history–future pair. These offsets equal ± 0.5, ± 1, and ± 2 bins of the 256-point future FFT. All models and scales remain frozen. Within each seed, the six offsets receive equal weight; Table 1 reports the mean and sample SD across three seeds. Relative error changes are computed from unrounded means.

A.2 Normalization and preprocessing

ETTh1, ETTm1, and Weather use chronological splits. Following the ETT convention (Zhou et al., 2021), StandardScaler statistics are fitted on the training segment for each variable j. For a raw observation $u_{t,j}$, the standardized value is

$$x_{t,j} = \frac{u_{t,j} - \mu_j^{\mathrm{tr}}}{\sigma_j^{\mathrm{tr}}}, \qquad \mu_j^{\mathrm{tr}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} u_{t,j}, \qquad (\sigma_j^{\mathrm{tr}})^2 = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} (u_{t,j} - \mu_j^{\mathrm{tr}})^2. \tag{11}$$

Here $\mathcal{T}$ is the training segment. Each variable is then normalized using its $L = 96$ observed history values:

$$m_j = \frac{1}{L} \sum_{t=1}^{L} x_{t,j}, \qquad s_j = \sqrt{\frac{1}{L} \sum_{t=1}^{L} (x_{t,j} - m_j)^2 + 10^{-5}}, \qquad \widetilde{x}_{t,j} = \frac{x_{t,j} - m_j}{s_j}. \tag{12}$$

The compact head pads 16 zeros on each side to form a 128-point input. Its 128 predicted values are cropped to the central 96 and restored as $\widehat{y}_{t,j} = s_j \widehat{\widetilde{y}}_{t,j} + m_j$. Reported macroscopic errors are in StandardScaler coordinates after this instance-normalization reversal. The history mean and scale are also used for future restoration; future values do not enter their estimation. Complex I/Q histories use the RMS normalization and phase restoration in Equations (1) and (7).

A.3 ARCHITECTURE AND OPTIMIZATION

The compact heads use width 256. SpikF uses its official forward implementation with the task-specific configurations in Table 5. Both predictors use 16 simulation steps. Adam uses learning rate 5×10^{-4} and batch size 32 with its default moment coefficients $(\beta_1, \beta_2) = (0.9, 0.999)$. Cosine scheduling uses $T_{\max} = 20$.

Table 5: Prediction-head configurations and training budgets. ETT denotes ETTh1 and ETTm1.

Setting	Configuration
Compact I/Q head	$2049 \to 256 \to 512$
Compact real head	$257 \to 256 \to 128$
SpikF, NeLoRa	Patch number/width/hidden width $32/32/360$, one block
SpikF, ETT	Patch number/width/hidden width $48/32/720$, one block
SpikF, Weather	Patch number/width/hidden width $32/32/360$, two blocks
NeLoRa seeds	SpikF 0/1/2; compact-head initializations 100/101/102
NeLoRa training	15-epoch cap, patience four, validation-MAE selection
Chirp factorization	Same stopping rule; paired head initialization at seed 100
Cross-domain training	ETT SpikF: eight-epoch cap; width-256 heads and Weather: 15-epoch cap
Coding ablation	Three epochs per group; paired head seeds and identical training-derived scales

The compact LIF head uses leak 0.9, threshold 0.5, subtractive reset, and spike-count readout. The selector is frozen during head training, while its order output remains history dependent. DR-LTC uses 64 ticks, a boundary at 32, and a terminal complex-magnitude gate at $0.03U$; stage scales are calibrated from training histories and held fixed. The coding comparison's three head seeds do not resample the frozen frontend, so their variability measures head training rather than selector training.

A.4 HARDWARE DEPLOYMENT AND MEASUREMENT PROTOCOL

Models and execution boundary. Both device comparisons use historical single-seed NeLoRa checkpoints and the same 2048 development windows, not the three-seed forecasting models or a new accuracy benchmark. SpikF retains its original 32 patches, patch width 32, hidden width 360, one spiking frequency selector (SFS) block, and 16 simulation steps. Its cached input contains the final SFS output at all 16 steps. The deployed decoder executes dense1, batch normalization, a Gaussian error linear unit (GELU), dense2, and temporal averaging; the host applies the original per-variable mean and standard-deviation restoration. The width-128 and width-256 SNN-FrFT heads reuse the original compiled weights and execute an input projection, 16 hard-LIF steps, and a readout, followed by host phase/RMS restoration. Each head forms one accelerator subgraph without central processing unit (CPU) fallback for its head operators. No model was retrained or pruned for this comparison.

The packed SpikF input has 32,772 float32 values per window (131,088 bytes), including four normalization values; SNN-FrFT uses 2050 float32 values (8200 bytes). Transfers preserve these native interfaces rather than padding them to equal sizes. Thus the host-inclusive comparison measures the deployed heads and their interfaces, not isolated kernels. Both frontends are excluded.

Board and sampling configuration. HM100 runs at 400 MHz, using Lyngor 1.23.0 with the KA200 target and synchronous lynpy calls. Board power is read passively with `mcutools -m samples`. Board-level and module-level sensor readings are recorded separately; Table 3 uses board-level energy. The sensor reports low-rate samples with 0.1 W display resolution. Each model is warmed up for at least 128 windows; an initial 2048-window pass checks outputs. Each power measurement brackets at least 20 seconds of load with eight seconds of idle measurement before and after, ending at a complete 2048-window traversal boundary.

The three measurement rounds use the orders SpikF/128/256, 128/256/SpikF, and 256/SpikF/128. These are repeated hardware measurements of fixed models, not training-seed replicates. The mean and sample SD are computed across the three runs for each head. The two SNN-FrFT widths differ by less than the observed energy variation, so the comparison does not establish an intrinsic energy ordering between widths.

Energy and throughput. For a load interval of duration T containing W processed windows, board power $P(t)$ is integrated by the trapezoidal rule. Let $\overline{P}_{\mathrm{idle}}$ be the average of the before/after idle means. Dynamic and total energy per window are

$$E_{\mathrm{dyn}} = \frac{\int_0^T P(t)\,dt - \overline{P}_{\mathrm{idle}} T}{W}, \qquad E_{\mathrm{total}} = \frac{\int_0^T P(t)\,dt}{W}. \tag{13}$$

Negative baseline-subtracted differences are not clipped. Sustained throughput on both devices is W/T, including input transfers, synchronous device execution, output transfers, and the original host restoration. Energy is measured at the board sensor, not at the host-machine wall outlet. Reported energy ratios divide measured means, rather than averaging per-run ratios; three measurement repeats do not establish statistical significance.

GTX 1650 measurements. All three cached heads are measured on GTX 1650 with the same weights, packed inputs, batch size, host restoration, and three-round interleaving as the HM100 study. GPU inference uses PyTorch 2.7.1, CUDA 11.8, and 32-bit floating-point (FP32) arithmetic. After 128 warm-up windows, the first full 2048-window traversal is compared with each model's saved FP32 predictions; all three pass the joint output criterion below. Each load lasts at least 20 seconds and ends at a full traversal boundary. Energy and throughput use the same load interval. The NVIDIA Management Library (NVML) reads GPU-board power at a target 10 ms polling interval, which does not specify the physical sensor refresh rate. CPU and whole-system power are excluded.

Before and after each load, a 20-second cooldown precedes the first stable five-second idle interval. Stability requires power SD at most 0.5 W, range at most 1.5 W, and first-to-last-second mean difference at most 0.5 W. Dynamic energy subtracts the average of the two time-integrated idle powers. All nine measurements pass both idle checks and the prespecified 1 W bound on before/after idle-mean drift. Every run is retained. Table 3 uses these three-repeat GPU measurements in place of the historical single-run references. Cross-device ratios compare board sensors and deployment runtimes under the same cached-head boundary.

Compiled-output diagnostics. The predefined joint numerical-equivalence criterion requires maximum absolute error at most 10^{-4} and normalized root-mean-square error (NRMSE) at most 10^{-3} relative to software FP32 outputs. None of the three HM100 heads meets both thresholds (Table 6). SpikF meets the NRMSE threshold but exceeds the maximum-error threshold; both SNN-FrFT heads exceed both. Task MAE is close to each model's software value, but this does not imply pointwise equivalence or equal predictive accuracy across models. These are compiled-deployment diagnostics on historical development windows.

Table 6: Numerical diagnostics for compiled HM100 heads on the historical 2048-window development set. Task MAE averages real and imaginary components.

Head	Max. absolute error	NRMSE (%)	FP32 MAE	HM100 MAE
SpikF	1.419894×10^{-4}	0.0807	0.00244391	0.00244313
SNN-FrFT (128)	1.060043×10^{-3}	0.3823	0.00957266	0.00957253
SNN-FrFT (256)	8.361097×10^{-4}	0.3457	0.00798814	0.00798771

B Numerical Values for Plotted Comparisons

Tables 7 and 8 retain numerical values and additional metrics underlying Figures 4 and 6. They use the same archived runs as the main-text figures; the cached-head energy table is reported in the main text in Section 4.6.

Table 7: Synthetic Chirp representation–restoration factorization on 1024 test windows. The first four rows form the matched 2×2 experiment, and the final row is a time-domain control. All heads have width 256.

Variant	MAE ↓	MSE ↓	History k_{90}	Peak (%)
Fixed FFT, no restoration	0.645774	0.542509	257.397	4.49
Fixed FFT + restoration	0.647458	0.545423	257.397	5.66
SNN-FrFT, no restoration	0.642669	0.528945	12.266	0.49
SNN-FrFT	**0.514547**	**0.444196**	12.266	17.19
Time-domain head	0.650129	0.555158	893.242	5.57

Bold denotes the best matched 2×2 entry. The time-domain row counts time samples for k_{90}.

Table 8: Numerical values for Figure 6. NeLoRa has 1015 windows and 2048 terminal rails per window. Prediction metrics are mean $\pm$ sample SD over three head seeds after three epochs. Only full DR-LTC uses the $0.03U$ terminal gate; gate-off retains its preceding stages. Single-region log uses the LP-TTFS response family within the radix-2 frontend, rather than the full PTSN/radix-4 method of Wang et al. (2026).

Coding	Events/window	Silent (%)	SF10 peak (%)	Error (bins)
Linear TTFS	331.6	83.81	70.47 ± 0.46	0.3195 ± 0.0302
DR-LTC, gate off	1111.4	45.73	75.08 ± 1.59	0.0995 ± 0.0062
DR-LTC	531.2	74.06	74.47 ± 1.67	0.0992 ± 0.0060
Single-region log (LP-TTFS response)	1023.8	50.01	75.18 ± 1.51	0.0985 ± 0.0062