FlexSpy: Side-Channel Spy Framework for Flexible Spiking Neuromorphic Hardware

Brojogopal Sapui*, Priyanjana Pal* and Mehdi B. Tahoori
Karlsruhe Institute of Technology, Germany
{brojogopal.sapui, priyanjana.pal, mehdi.tahoori}@kit.edu

***Abstract*—Biologically-inspired Spiking Neural Networks (SNNs) are promising for energy-efficient neuromorphic computing in edge applications such as soft robotics, wearable health monitors, and IoT devices. Recent advances in flexible electronics enable deployment on thin, ultralow-cost conformal substrates. However, reduced packaging and thin encapsulation increase exposure to power side-channel leakage, while limited I/O and shared supply rails restrict conventional split-domain defenses. In this work, we present *FlexSpy*, a side-channel framework that integrates spike-accurate transient simulation with correlation and mutual information (MI) analysis, and introduces a Spike-Leakage Index (SLI) to localize vulnerable design blocks. Rather than targeting per-synapse weight extraction, the framework enables network-level recovery, label inference, and layer-wise spike-rate estimation. As a baseline, we compare against flexible recurrent neural networks (*f*-RNNs), which exhibit weaker leakage due to smoother recurrent dynamics. Across operating corners, *FlexSpy* achieves ROC-AUC $= 0.91$ and recovers layer-wise spike rates with $\mathrm{NMSE} = 0.14$ and cosine similarity $= 0.93$. As countermeasure, spike-time randomization and event balancing are proposed to reduce leakage by 38–70% at $\leq 9\%$ power overhead.**

I. Introduction

Modern embedded intelligence increasingly relies on neuromorphic computing for ultra-low-power cognitive processing in applications such as wearable health monitoring, edge intelligence, and large-scale IoT. These domains require platforms that are flexible, lightweight, and robust while enabling on-device computation. However, conventional silicon is constrained by rigid substrates, high-temperature fabrication, and costly infrastructure [1]. As an alternative, flexible electronics (FE) based on amorphous indium-gallium-zinc oxide (a-IGZO) thin-film transistors (TFTs) offer low-temperature fabrication, high mobility, and low leakage on flexible or ultrathin substrates, enabling bendable and transparent circuit integration.

Among various computing paradigms suited to flexible hardware [2]–[4], spiking neural networks (SNNs) [5] have emerged as a highly promising alternative toward energy-efficient neuromorphic computing. By emulating the event-driven communication of biological neurons, flexible spiking neural networks (f-SNNs) perform sparse, asynchronous computation through discrete electrical spikes, thereby eliminating redundant switching activity common in artificial neural networks (ANNs) [6, 7]. When implemented using analog circuitry, f-SNNs naturally leverage the continuous-time dynamics of TFTs to achieve real-time processing with minimal power making it attractive for bio-conformal, e-skin, and edge artificial intelligence (AI).

However, the same attributes that make FE attractive introduce security exposure that has not been well studied so far. Flexible substrates typically lack protective packaging, metallic shielding, and multi-layer ground planes; combined with limited I/O, shared supply rails, and thin dielectrics, which increase susceptibility to power side-channel leakage. In f-SNNs, small variations in spike timing, amplitude, or rate can reveal neuron activity patterns or learned parameters. As f-SNNs compute in *analog* domain, leakage arises from continuous-time device/circuit dynamics (e.g., spike-waveform and substrate coupling) and is fundamentally different from digital logic, standard AI accelerators; and other well-studied threat models. Understanding these analog leak paths is the *first step* toward security-aware design of in/near-sensor AI.

To address this gap, we propose *FlexSpy*, a *side-channel spy framework* for flexible spiking neuromorphic hardware which couples spike-accurate transient simulation to model data-dependent power leakage in neuron and synapse circuits. The adversary inserts a small series shunt on V_{DD} and records a single supply trace $I_{DD}(t)$ during normal operation without internal probing. For spike-time alignment we introduce a *virtual trigger* $V_{\text{trig}}(t)$, a derived signal from the same trace to segment spike epochs. Throughout, we focus on *design-level* recovery of labels, regressing layer-wise spike-rate vectors, and profiling coarse structure (e.g., synapse multiplicity), rather than per-synapse weight extraction from a single supply rail. The main contributions are:

- A flexible *substrate-aware* leakage model for f-SNNs explaining why aggregated synaptic conductance produces quasi-DC offsets and dominates observable leakage.
- A *trace-synthesis and analysis* pipeline that links process-design kit (PDK) simulations to spike-aligned features and calibrated attacks for early, repeatable security assessment.
- Spike-time randomization and event balancing are proposed to reduce leakage with minimum overhead.

Experimental analysis on a-IGZO f-SNNs shows robust leakage: ROC–AUC 0.91 at nominal (to $0.85/0.82$ at low-V_{DD}/high-T) and recoverable layer-wise rates from power ($\mathrm{NMSE} = 0.14$, cosine $= 0.93$). Applying the proposed countermeasures reduces measurable leakage by 38–70% at $\leq 9\%$ power overhead with negligible accuracy loss. As a reference baseline, we also benchmark flexible recurrent neural network (f-RNN), which exhibits weaker per-window leakage and higher recovery error due to smoother recurrent dynamics.

The rest of the paper is structured as follows. Sec. II reviews a-IGZO-based FE, side-channel attack (SCA), and related work on neuromorphic circuits. Sec. III discusses the *FlexSpy* method. Sec. IV reports experimental results and mitigation efficacy. Sec. V concludes our work with future directions.

II. Preliminaries

A. Flexible Electronics (FE)

FE offer mechanical flexibility, lightweight, and compatibility with substrates such as plastics and ultrathin glass,

*Authors contributed equally to this work.

allowing conformal electronics for wearables and smart packaging [1]. At the device level, FE typically employ only N-type a-IGZO TFTs fabricated at low temperature, which supports cost-effective, large-area processing on temperature-limited substrates. A typical flow prepares a rigid carrier, casts and cures a flexible film, deposits and patterns the a-IGZO channel (e.g., by ALD or PVD), and completes the dielectric and metallization stack.

B. Flexible Neural Networks (f-NNs)

Analog flexible neural networks (f-NNs) implement conductance-weighted sums with device nonlinearities and RC dynamics [2, 8, 9]. On a-IGZO TFTs, this enables efficient continuous-time primitives approximated by simple RC behavior. Two common realizations are event-driven spiking (f-SNNs) and continuous-time recurrent f-NNs (f-RNNs).

1) Flexible Spiking Neural Networks (f-SNNs): Spiking neurons [5, 8] communicate via discrete events and remain idle between spikes. As shown in Fig. 1 (left), a typical f-SNN cell integrates (i) a synaptic stage forming $i_{\mathrm{syn}}(t) = \sum_i w_i s_i(t)$, (ii) an RC integrator accumulating charge to threshold, and (iii) reset control enforcing refractoriness. These blocks map naturally to TFTs physics and support low-power operation on flexible substrates. However, continuous conduction paths and shared supplies induce data-dependent current variations, appearing as quasi-DC offsets during spikes with sharp transitions at spike edges, which later aid alignment and leakage extraction.

2) Flexible Recurrent Neural Networks (f-RNNs): Flexible recurrent designs [2] realize smooth state evolution without discrete spikes. Hidden states are stored on capacitors, while first- or second-order RC define learnable time constants τ_i=R_iC_i. Weighted sums and activations use resistive and transconductance stages, forming a continuous-time approximation of recurrent models, as illustrated in Fig. 1. Compared to event-driven f-SNNs, f-RNNs generate smoother supply-current profiles with weaker alignment features.

C. Side-Channel Analysis (SCA)

Side-channel analysis (SCA) extracts hidden information from physical leakages such as supply current, EM emissions, and timing. Attacks are broadly categorized as *non-profiled* (e.g., CPA/DPA) that correlate predicted leakage with measured traces, and *profiled* (template) [10] methods that learn statistical models from reference traces. Practical pipelines apply baseline correction, light filtering, and trace alignment (external or self-derived triggers), then analyze short windows to localize peak leakage. Beyond correlation and templates, regression enables recovery of continuous values, while mutual-information (MI) [11] provides model-independent leakage quantification. In event-driven neuromorphic systems, spike-induced low-frequency envelopes and sharp transitions make windowed analysis particularly effective.

D. Related Work

Flexible neuromorphic computing has progressed from single a-IGZO TFT neuron to fully analog networks and temporal processors. Recent f-RNN-style designs [2] employ second-order filters with variation-aware training to improve robustness under device and sensor noise [2]. On rigid silicon, numerous studies show exploitable power/EM leakages in neural accelerators, from early correlation attacks on CNN engines [12] to broader surveys of deep-learning SCA [13, 14] and model/parameter extraction from traces [15, 16]. For biologically inspired SNNs, *SCANN* demonstrated that CMOS SNNs leak spike activity and parameters via power signatures [17]. Compared with CMOS, flexible substrates amplify observability due to reduced shielding, thinner dielectrics, and shared rails. Beyond spiking/recurrent circuits, flexible bespoke MLP classifiers, also exhibit exploitable power side channels, via correlation-power analysis (CPA) (digital) and CNN-based profiling (analog) [3]. Unlike CMOS SNN, this paper contributes the first device-to-design leakage model and a unified early-EDA framework for f-SNN side-channel evaluation, expose spike-epoch quasi-DC offsets due to continuous conductance modulation on shared rails, with a comparison to state-of-the-art f-RNNs.

III. Proposed Method: *FlexSpy*

FlexSpy, as shown in Fig. 3, is a design-time framework for f-NCs that predicts, localizes, and mitigates power side-channel leakage before fabrication. Starting from a circuit netlist (as shown in Fig. 1) and technology-calibrated device models (with corner conditions). It (i) synthesizes time-accurate power traces under realistic power-delivery network (PDN) conditions, (ii) builds spike-aligned features that match the physics of a-IGZO TFTs, and (iii) applies a calibrated attack suite to quantify what leaks, when it leaks, and from which blocks it leaks. Finally, it evaluates countermeasures in-loop, reporting leakage reduction and power estimates, so the design trade-offs are visible early.

A. Threat Model and Measured Signals

We assume a non-invasive adversary with access during normal operation. The attacker inserts a small shunt (1-10 Ω) in series with V_{DD} and records a single power trace $I_{\mathrm{DD}}(t)$. Because spike-epoch timing matters for alignment, we use a *virtual trigger* signal, denoted $V_{trig}(t)$, which is *derived from the same shunt-based measurement* and is used only for spike-window segmentation.

B. Flexible Substrate-Aware Leakage Model

Flexible a-IGZO TFTs are n-type unipolar and rely mainly on resistive biasing. During spike epochs, the dominant *data-dependent* current flows quasi-statically through the enabled synaptic paths. Short displacement currents add smaller transients. We use the following model:

$$I_{DD}(t) \;\approx\; I_{\text{bias}} + V_{\text{he}} \sum_i g_i \, s_i(t) + \sum_k C_k \, \frac{dV_k}{dt} + n(t). \quad (1)$$

where I_{bias} is idle current, g_i are synaptic conductances, $s_i(t)$ indicates synaptic activity (rate/latency), V_{he} is neuron head-node potential during a spike, V_k are internal node voltages associated with displacement currents, and $n(t)$ represents device and instrumentation noise. At ms/kHz scales, $V_{\text{he}} \sum_i g_i s_i(t)$ produces sustained level shifts (*quasi-DC offsets*) that are easily observed after baseline removal. This behavior is strong in f-SNNs, where spikes activate discrete conductance bundles, and is weaker, but still present, in f-RNNs, where hidden states evolve smoothly and produce lower envelopes.

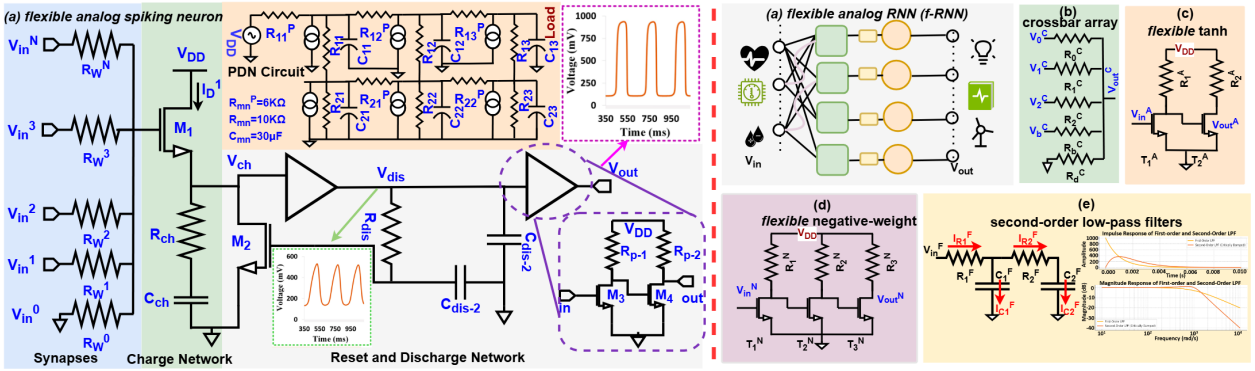
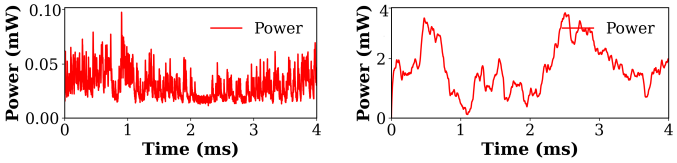


Fig. 1: Circuit-level implementations of flexible analog neuromorphic circuits (f-NCs). Left: f-SNN cell with Synapse, Charge, and Reset/Discharge stages. Right: f-RNN cell with recurrent RC dynamics.



(a) Traces of f-SNN showing spike-triggered current patterns.

(b) Traces of f-RNNs showing continuous oscillations.

Fig. 2: Distinct leakage from raw power traces: spike-driven offsets in f-SNN vs. smooth RC oscillations in f-RNN.

C. Trace Synthesis

Neuron and synapse blocks are instantiated using a-IGZO TFTs, poly-resistors, and MIM capacitors. A compact PDN model (series rail resistance with on-board decoupling) captures finite sheet resistance and discrete capacitors typical of flexible backplanes. Transient simulations run in ms-long windows across supply and temperature corners. For each stimulus, *FlexSpy* records $I_{DD}(t)$ at the PDN output. Small input jitter and Gaussian noise is added to explore the measurements.

D. Spike-Aligned Feature Extraction

The processing of traces is carried out in three steps. First, the idle baseline I_{bias} is estimated from quiescent segments and subtracted to reveal dc-level shifts. Second, light low-/band-pass filtering preserves ms-scale dynamics while suppressing measurement noise. Third, spike epochs $[t_0, t_1]$ are segmented using edges in a virtual trigger signal. We compute a *virtual trigger* signal, from the shunt-based power trace $I_{\text{DD}}(t)$ as:

$$V_{\text{trig}}(t) = \alpha \frac{dI_{\text{DD}}(t)}{dt} \approx \frac{dV_{\text{shunt}}(t)}{dt}, \quad V_{\text{shunt}} = R_s I_{\text{DD}}. \quad (2)$$

followed by light smoothing to suppress high-frequency noise. $V_{trig}(t)$ highlights spike onsets/offsets and is used solely to segment spike epochs without any internal trigger.

Windowing. We perform edge-to-edge windowing on $V_{trig}(t)$: detect rising/falling edges by thresholding $V_{\text{trig}}(t)$ at $\pm\theta$ with a minimum inter-event gap $\Delta t_{\min}$, then define spike windows $[t_0, t_1]$ around consecutive edge pairs (with small padding). This produces stable windows under baseline drift and across PVT corners.

Per-window features. For each window w we compute per-corner (i) the quasi-DC level:

$$\Delta I_{\text{DC}}^{(w)} = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (I_{\text{DD}}(t) - I_{\text{bias}})\, dt. \quad (3)$$

(ii) a spike-count estimate from $V_{trig}(t)$, (iii) inter-spike interval (ISI) statistics, and (iv) optional timing context.

E. Attack Suite: f-SNNs vs. f-RNNs

After feature extraction, the attack suite is described in algorithm 1 and the subsequent paragraphs expand each stage of the Algorithm: CPA, template profiling, regression, and MI/SLI.

Algorithm 1 FLEXSPY FRAMEWORK: end-to-end procedure

Require: Netlist $\mathcal{N}$; device models with corner settings $\mathcal{M}$; stimuli $\mathcal{X}$; analysis targets $\mathcal{O}$; PDN parameters

Ensure: Recovered parameters: (i) dominant leakage window $w^\star$ and its features $Z^{(w^\star)}$; (ii) input/class hypothesis $\hat{y}$ (or $\hat{x}$) with posterior $p(\hat{y} \mid \mathcal{Z})$; (iii) per-layer rate vectors $\hat{\mathbf{r}}^{(\ell)}$ for f-SNNs or hidden-state estimates $\hat{\mathbf{h}}_t^{(\ell)}$ for f-RNNs; (iv) structural targets $\hat{k}$ (synapse multiplicity) and $\hat{c}$ (spiking source cluster); (v) SLI-based hotspot ranking $\mathcal{H}_{\text{SLI}}$.

1: **Synthesize traces:** for each $(x, \text{corner}) \in \mathcal{X} \times \mathcal{M}$, run transients on $\mathcal{N}$ (+PDN); record $I_{\text{DD}}(t)$; compute $V_{\text{trig}}(t)$

2: **Build features:** baseline removal $\rightarrow$ light filtering $\rightarrow$ windowing from $V_{\text{trig}}(t) \rightarrow$ per-window features $\mathcal{Z} = \{\Delta I_{\text{DC}}, \text{count}, \text{ISI}, \text{timing}\}$; standardize by corner

3: **CPA (localize + shortlist):** compute windowwise correlations between $\mathcal{Z}$ and $x = \sum_i g_i \bar{s}_i$; **Output:** $w^\star = \arg\max_w \rho_w$ (leakage window) and a hypothesis $\mathcal{H}$ (candidates for input bins in f-SNN or hidden-state in f-RNN)

4: **Profiling (discrete recovery):** fit Gaussian (diag/LDA/QDA) templates for targets in $\mathcal{O}$ (e.g., class y, multiplicity k, source cluster c); **Output:** $\hat{y}$ (class label), $\hat{k}$, $\hat{c}$ with posteriors $p(\cdot \mid \mathcal{Z})$

5: **Regression (continuous recovery):** train ridge models $\mathcal{Z} \mapsto \mathbf{r}^{(\ell)}$ (f-SNNs) or $\mathcal{Z} \mapsto \mathbf{h}_t^{(\ell)}$ (f-RNNs); **Output:** $\hat{\mathbf{r}}^{(\ell)}$ or $\hat{\mathbf{h}}_t^{(\ell)}$.

6: **MI/SLI (attribution + hotspots):** estimate design/device attribution and compute $\text{SLI}(b, w; S)$; **Output:** ranked hotspot set $\mathcal{H}_{\text{SLI}}$ and attribution map (device vs. design).

7: **Countermeasures (configuration recovery):** sweep jitter, balancing, and PDN; re-run steps 1–6; **Output:** configuration achieving target leakage $\leq \tau$ with reported overheads.

Correlation Power Analysis (CPA): CPA measures how strongly the trace aligns with a linear predictor of rate-weighted activity $x = \sum_i g_i \bar{s}_i$. Using a derivative-based *virtual trigger* to align spike windows, sliding CPA localizes the time of maximum leakage: in f-SNNs it yields sharp peaks inside spike epochs, while in f-RNNs peaks are broader and smaller due to

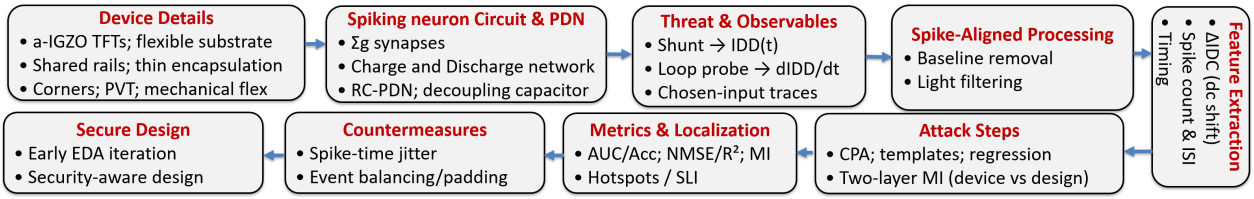


Fig. 3: Overview of *FlexSpy*. From device models and PDN details, the pipeline performs spike-accurate transient simulation, spike-aligned feature extraction, and calibrated SCA (CPA, profiling, regression, MI). Leakage metrics and a Spike-Leakage Index (SLI) guide evaluation of countermeasures, enabling security-aware, early-stage design of f-NCs.

smoother state evolution. It confirms quasi-DC dominance in the most revealing window.

During a spike window, the subset of active synapses conducts simultaneously, producing a quasi-DC elevation in the supply current; the resulting ΔI_{DC} offset is approximately proportional to the summed conductance $\sum_i g_i$ and thus encodes rate-weighted synaptic activity. Different inputs excite different synapse subsets, yielding distinct ΔI_{DC} signatures; sliding CPA tests input hypotheses by correlating the measured offset with predicted activity $x = \sum_i g_i\, \bar{s}_i$, and only the correct hypothesis reproduces the dominant offset, giving a sharp peak in the corresponding window.

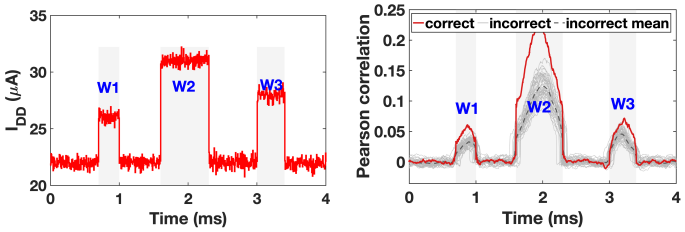
1) Template Profiling (Gaussian).: Beyond identifying the correct input, the next goal is to recover finer internal quantities such as layer-wise synaptic rates, neuron activation profiles, or continuous conductance trends that are not fully captured by a linear CPA model. Template attacks address this by learning the full multivariate distribution of ΔI_{DC} features for each class or internal state, enabling high-resolution classification and layer-rate recovery even under noise. *FlexSpy* trains Gaussian templates on concatenated window features for hypotheses such as class y, synapse multiplicity k, or source cluster c. By default we use diagonal or LDA covariances with shrinkage (stable with limited traces). When the per-class variance differs significantly, we switch to QDA or small mixtures. At attack time, per-window log-likelihoods are summed, which is robust to modest noise and corner drift. In f-SNNs, templates exploit distinct offset levels across windows. In f-RNNs, they leverage lower-frequency envelopes and timing. Next, Regression and MI analysis go further: regression infers continuous latent variables directly from trace amplitudes, while MI quantifies how much information each window or feature leaks.

2) Regression for Continuous Recovery: Ridge regressors map window features to per-layer rate vectors $\mathbf{r}^{(\ell)}$ (for f-SNNs) or to hidden-state estimates for f-RNNs. High R^2 and low NMSE indicate that windowed ΔI_{DC} retains rate-weighted activity. Here, f-SNN rates are recovered more accurately than f-RNN hidden states, as SNN's offsets are closer to a linear readout of $\sum_i g_i \bar{s}_i$.

3) Mutual Information.: We quantify the share of information attributable to device-specific variations versus design-level activity. Let L_{device} denote features that capture device variation (e.g., idle $1/f$ slope, narrowband lines) and L_{design} denote spike-window observables (e.g., ΔI_{DC}, counts, ISI). For a secret S (labels y, rates $\mathbf{r}^{(\ell)}$, multiplicity k, cluster c):

$$I(S; L_{\text{device}}, L_{\text{design}}) = I(S; L_{\text{device}}) + I(S; L_{\text{design}} \mid L_{\text{device}}) \quad (4)$$

Histogram or kNN estimators with bootstrap confidence intervals provide stable estimates. In our f-SNN experiments, the conditional term dominates. In f-RNNs it still carries most information, but the gap narrows due to smoother dynamics.



(a) Measured supply current $I_{DD}(t)$ with shaded spike windows (W1-W3). Quasi-DC offsets appear during spike epochs in f-SNNs.

(b) Sliding-CPA time localization. A bundle of incorrect hypotheses (gray) stays near 0 except in spike windows; correct hypothesis (red) peaks in *W2*.

Fig. 4: $I_{DD}(t)$ trace shows the quasi-DC ΔI_{DC} offsets that drive it (left), whereas correlation peaks align with spike-epoch current offsets in agreement with Eq. 1, sliding-CPA shows *when* leakage is maximal (right) with P-CONS dataset.

4) Spike-Leakage Index (SLI) for Hotspot: Side-channel leakage in flexible neuromorphic circuits arises from both *where* the computation occurs (circuit block) and *when* it occurs (spike-aligned window). To quantify these, we define an SLI that measures how much information a specific pair reveals about a secret S.

Let b index a functional circuit block (e.g., Synapse, Charge, and Reset/Discharge stages in f-SNNs; recurrent RC filters or state capacitors in f-RNNs), and let w denote a spike-aligned time window. For each (b, w), *FlexSpy* extracts the corresponding feature vector $Z_b^{(w)}$. The SLI is defined as:

$$\mathrm{SLI}(b, w; S) = \frac{I(S; Z_b^{(w)})}{H(S)} \in [0, 1]. \quad (5)$$

where $I(S; Z_b^{(w)})$ is MI between the secret S and the features emitted at block b in window w, and $H(S)$ is entropy of S.

A high SLI value indicates that observing the activity of block b during window w reveals a large fraction of the total information about S. Thus, SLI jointly attributes leakage to both circuit structure (which block) and temporal dynamics (which window). This allows *FlexSpy* to rank vulnerability hotspots, identify computations that dominate information exposure, and determine where countermeasures will have the greatest impact.

To separate structural leakage from device-specific variation, *FlexSpy* also computes a *design-only* variant by removing device-variation components from $Z_b^{(w)}$. The resulting SLI maps provide a compact, interpretable summary of how information is distributed across blocks and spike epochs.

F. Countermeasures

1) Spike-time randomization.: We introduce bounded jitter ($\pm 5\%$ by default) at the neuron's reset path so that spike epochs no longer align across traces. In a-IGZO, this can be realized by modulating the effective threshold with a small auxiliary

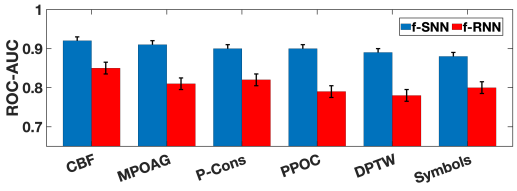


Fig. 5: Leakage at the nominal corner for benchmark datasets: ROC-AUC of label inference using spike-window features. Blue: f-SNN; red: f-RNN. Error bars show that f-SNN consistently leaks more than f-RNN.

current drawn from a compact pseudo-random source (e.g., a ring-oscillator-gated switch that periodically charges a noise-sampling capacitor). Jitter reduces the peak correlation inside a window and spreads energy in virtual trigger estimate $V_{\text{trig}}(t)$, lowering CPA peaks and weakening template alignment

2) Event balancing / dummy conductance.: To flatten the ΔI_{DC} offsets, we add a balancing branch that sources a small counter-current whenever a synapse is active. At the synapse cell, this is a switched resistor (a TFT acting as a resistor) connected from the neuron head node to ground (or to a complementary rail) through a gate driven by the same pre-synaptic event. Calibration is performed once per neuron: for each synapse i, a small binary-weighted resistor bank (e.g., $R, 2R, 4R$) is trimmed so that the incremental current δI_i offsets a fixed fraction of the synaptic current $g_i \cdot V_{\text{head}}$. Full flattening is unnecessary, partial balancing already reduces separability by pushing class clusters closer in ΔI_{DC}.

IV. Evaluation

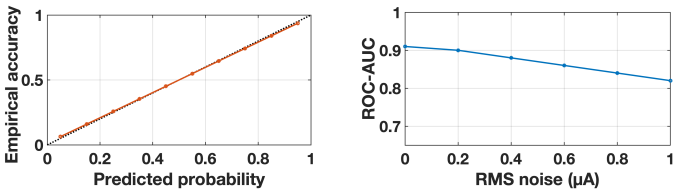
During evaluation, we quantify (i) design-level leakage induced by spike-driven activity and (ii) the effect of lightweight countermeasures. Transient simulations ran in *Cadence Virtuoso* with calibrated device models to capture power traces. The traces were analyzed by the Python pipeline (algorithm 1).

A. Experimental Setup

We implement f-SNN (synaptic input, integrate, reset/discharge) using n-type TFTs, poly-resistors, and MIM capacitors for each specific application targets. A PDN ladder (series rail resistance plus on-board decoupling, e.g., R^{P}_{mn}=6 kΩ, R_{mn}=10 kΩ, C_{mn}=30 μF) models the flexible large-area substrate. As leakage is extracted as a window-averaged quasi-DC offset, high-frequency measurement noise is attenuated, while contact, packaging, and bending effects are captured as PDN/corner perturbations without eliminating relative class separability. Transient simulation covers ms-length windows in the 1-5 kHz regime (1-5 μs steps) at nominal and two supply corners (V_{DD}=1.0 V, 3.0 V) over -20°C to 85°C. Each configuration yields ∼10 000 traces per stimulus. We record $I_{DD}(t)$ and derive $V_{trig}(t) \propto dI_{DD}/dt$ for alignment. After baseline removal and light low-pass filtering, spike epochs are segmented from $V_{trig}(t)$ edges. Per-window features include ΔI_{DC}, a spike-count estimate, and ISI statistics per corner. The input benchmark datasets are from *time-series UCR [18]* with a 70/15/15 split (profiling/validation/attack).

B. Results and Discussion

Label Inference from Power Leakage: A logistic classifier trained on the spike-window design feature vector $\mathbf{z}_{\text{design}}$ (Sec. III) achieves ROC–AUC of 0.91 at the nominal corner



(a) The model is slightly under-confident with ECE $< 5\%$; adding simple device-variation indicators yields only a minor calibration benefit in f-SNNs.

(b) Smooth AUC roll-off under added measurement noise (normalized), consistent with the dominance of spike-window observables over device variation.

Fig. 6: Reliability and measurement robustness in f-SNNs.

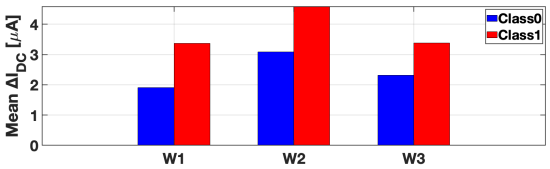


Fig. 7: Windowed current shift by class on f-SNN (P-Cons) shows that distinct quasi-DC offset reflect $\sum_i g_i s_i$ model.

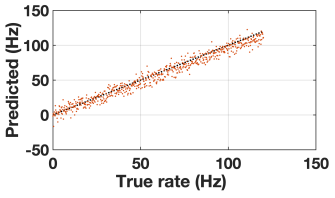
with ECE $< 5\%$. At low V_{DD} and at high T the AUC falls to 0.85 and 0.82, and adding simple device-variation indicators improves AUC by only about 0.02. Sliding-CPA localizes when leakage is maximal, with peaks confined to the dominant spike window (W2), which is consistent with the quasi-DC term in Eq. (1) and highlights that ΔI_{DC} drives the measurable signal.

Fig. 4 localizes when leakage occurs: sliding CPA peaks appear only within W2, consistent with Eq. 1. Cross-dataset results show the same pattern (Fig. 5), and the calibration/noise-stress plots indicate stable reliability with smooth AUC roll-off (Fig. 6). On P-Cons, class-dependent shifts in ΔI_{DC} (Fig. 7) confirm that the rate-weighted $\sum_i g_i s_i$ term dominates-making a timing-aligned power attack practical and robust.

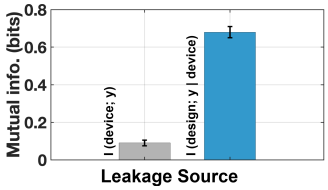
Spike-Rate Recovery: Ridge regression maps $\mathbf{z}_{\text{design}}$ to layer-wise firing-rate vectors $\mathbf{r}^{(\ell)}$. Across corners, NMSE $=$ 0.14 and cosine $=$ 0.93. Fig. 8(a) shows predicted vs. true rates; the near-linear trend reflects the static $\sum_i g_i \bar{s}_i$ term in Eq. 1 captured by ΔI_{DC}. Thus, an attacker can estimate layer-wise firing rates from power alone with low error, exposing intermediate activity.

Two-Layer MI: As in Fig. 8(b), we decompose device vs. design-level contributions via Eq. 4. For labels, $I(y; L_{\text{device}}) \approx$ 0.09 bits and $I(y; L_{\text{design}} \mid L_{\text{device}}) \approx$ 0.68 bits, so ∼88% of class information stems from spike-window features. For rates and multiplicity, conditional MI ranges 0.72-0.81 bits at nominal and drops by ∼20% at low-V_{DD}. Therefore, most recoverable information comes from design-level spike windows rather than device fingerprints.

Profiling Beyond Linear CPA: Gaussian templates trained on concatenated spike-window features recover richer structure than CPA. We evaluate two design-level targets that arise in f-NCs: (i) synapse *multiplicity* k (the number of active conductance paths contributing to a spike window), and (ii) *source clusters* (C1-C4), which group synaptic inputs by connectivity or receptive-field origin. Both targets reflect circuit-level structure and allow us to measure how much FlexSpy reveals beyond coarse leakage timing (matrices are shown in Fig. 9). At nominal conditions, multiplicity ($k \in \{0, \ldots, 4\}$) reaches 87.3% accuracy (F1 85.9%, ECE 3.2%), with con-

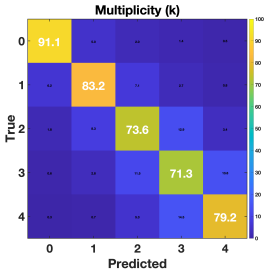


(a) Layer-wise rate recovery on held-out traces shows linearity (P-CONS); dashed line is identity.

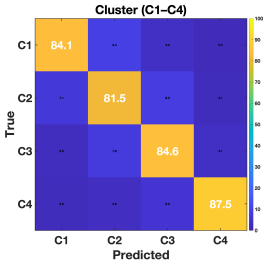


(b) Two-layer MI shows labels leak majorly via design-level spike-window activity (P-CONS).

Fig. 8: Rate recovery and mutual information in f-SNNs.



(a) Profiling (P-CONS) the number of active synaptic paths per spike window shows errors are mostly off-by-one (adjacent k).



(b) Source-cluster profiling (P-CONS) from spike-window features shows confusions concentrate on neighboring clusters.

Fig. 9: Confusion matrices for profiling from traces in f-SNNs. Gaussian templates trained on spike-window features recover synapse multiplicity and source cluster.

fusions concentrated on adjacent k values. Cluster inference (C1-C4) achieves 83.1% Top-1 / 95.2% Top-2 when including simple timing context. Corner degradation is modest (e.g., 82% / 79% at high-T/low-V_{DD}), confirming robustness of profiling. These results show that power traces leak not only labels but also structural attributes (multiplicity and source cluster).

Comparative Study off-SNN vs. f-RNN: To understand how flexible dynamical primitives differ in leakage behavior, we repeat the analysis for recurrent flexible circuits with second-order RC filters. The f-RNN produces low-frequency oscillatory envelopes, whereas the f-SNN exhibits quasi-DC steps aligned with spike epochs. Under identical rails and corners, f-SNN achieves higher AUC and lower rate-recovery NMSE due to stronger static-current coupling, while f-RNN hides instantaneous spikes but accumulates leakage over time. Results are summarized in Fig. 10.

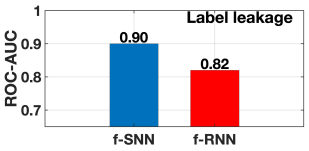
1) Countermeasure: Two lightweight countermeasures (bounded spike-time jitter and event balancing) are evaluated in-loop. Tab. I shows both suppress the dominant W2 hotspot by breaking alignment and flattening quasi-DC offsets, with the combination strongest; Tab. II indicates modest area/power cost and negligible accuracy impact, and gains are larger for f-SNN than f-RNN due to stronger spike-window ΔI_{DC}. Overall leakage drops up to 70% in f-SNN and about 22–30% in f-RNN-consistent across datasets (Tab. II).

TABLE I: SLI (dominant window W2) for f-SNN.

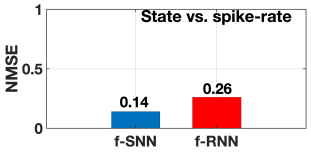
Block (P-CONS)	Baseline	+Jitter	+Balancing	+Both
Synapse (W2)	0.55	0.43	0.22	0.17
Charge/Integrate (W2)	0.40	0.31	0.16	0.12
Reset/Discharge (W2)	0.18	0.14	0.07	0.05

V. CONCLUSION

SNNs are promising for on-skin and conformal intelligence but remain vulnerable due to shared supplies and minimal



(a) Label leakage (ROC–AUC): f-SNN 0.90 $>$ f-RNN 0.82.



(b) State–rate NMSE: f-SNN 0.14 $<$ f-RNN 0.26.

Fig. 10: f-SNN (P-CONS) exhibits larger ΔI_{DC} offsets per window than f-RNN, producing stronger instantaneous leak.

TABLE II: Countermeasure overhead and leakage reduction

Dataset	f-SNN				f-RNN			
	Area↑ (%)	Power↑ (%)	Acc. Δ (pp)	Leak↓ (%)	Area↑ (%)	Power↑ (%)	Acc. Δ (pp)	Leak↓ (%)
CBF	3.5	7.0	−0.2%	60	2.8	5.5	−0.2%	29
P-Cons	4.0	8.0	−0.3%	56	3.0	6.0	−0.2%	24
PPOC	4.5	8.5	−0.4%	53	3.2	6.2	−0.3%	23
DPTW	5.0	9.0	−0.5%	50	3.5	6.5	−0.3%	22
MPOAG	4.0	8.0	−0.3%	55	3.0	6.0	−0.2%	25
Symbols	3.5	7.0	−0.2%	70	2.8	5.5	−0.2%	38

Numbers reflect medians across sweeps.

packaging. *FlexSpy* introduces a substrate-aware, design-time framework that links device physics, circuit dynamics, and design-level leakage through a unified device to design model. It enables early assessment of vulnerability and guided application of lightweight countermeasures during design. It offers a path toward security-aware development of f-NCs, with future directions including parasitic-aware modeling, EM-channel analysis, automated allocation of jitter resources, and co-design with training to secure f-SNNs.

VI. ACKNOWLEDGMENT

This work has been supported by the European Research Council (ERC) (Grant No. 101052764).

REFERENCES

[1] J. C. Costa *et al.*, "Flexible Sensors — From Materials to Applications," *Technologies*, vol. 7, 2019.
[2] T. Gheshlaghi *et al.*, "Adapt-pnc: Mitigating device variability and sensor noise in printed neuromorphic circuits with so adaptive learnable filters," IEEE, Tech. Rep., 2025.
[3] P. Pal *et al.*, "Side channel vulnerability analysis of flexible neuromorphic circuits," in *IEEE/ACM ICCAD*, 2025.
[4] E. Ozer *et al.*, "Bespoke machine learning processor development framework on flexible substrates," in *2019 IEEE FLEPS*, 2019, pp. 1–3.
[5] M. Lopez *et al.*, "A tunable multi-timescale indium-gallium-zinc-oxide thin-film transistor neuron towards hybrid solutions for spiking neuromorphic applications," *Communications Engineering*, vol. 3, 07 2024.
[6] P. Pal *et al.*, "Invited paper: Side channel vulnerability analysis of flexible neuromorphic circuits," in *2025 IEEE/ACM ICCAD*, 2025, pp. 1–7.
[7] ——, "Thermo-nas: Thermal-resilient ultralow-cost igzo-based flexible neuromorphic circuits," in *2026 31st ASP-DAC*, 2026, pp. 244–250.
[8] ——, "Analog printed spiking neuromorphic circuit," in *IEEE DATE*, 2024, p. 6 S.
[9] Z. Zhai *et al.*, "Temporal-rate encoding to realize unary positional representation in spiking neural systems," *CoRR*, vol. 2012.09912, 2020.
[10] S. Chari *et al.*, "Template attacks," in *CHES 2002*. Springer, pp. 13–28.
[11] N. Veyrat-Charvillon *et al.*, "Mutual information analysis: How, when and why?" in *CHES 2009*, vol. 5747. Springer, 2009, pp. 429–443.
[12] L. Wei *et al.*, "Power analysis attack and countermeasure for hardware neural networks," in *Proc. ACM AsiaCCS*, 2018, pp. 735–747.
[13] J. Méndez Real *et al.*, "Physical side-channel attacks on deep neural networks: A survey," *Applied Sciences*, vol. 11, p. 7369, 2021.
[14] B. Acsok *et al.*, "Sok: Deep learning–based physical side-channel analysis," *ACM Computing Surveys*, vol. 55, pp. 1–37, 2023.
[15] D. Horváth *et al.*, "Model extraction from deep neural networks via physical side channels," in *USENIX Security Symp.*, 2024, pp. 1153–1170.
[16] L. Yan *et al.*, "Mercury: Model extraction via remote side channels on dnn accelerators," in *Proc. IEEE FPT*, 2023, pp. 123–130.
[17] R. Nagarajan *et al.*, "Scann: Side-channel analysis of spiking neural networks," *Chips*, vol. 3, pp. 335–349, 2023.
[18] Y. Chen *et al.*, "The ucr time series classification archive," July 2015, www.cs.ucr.edu/~eamonn/time_series_data/.