Practical Mechanism for Fault-Tolerant Spiking Neural Networks

via Simple Input Control Based on Learnable Fragmentation

Hyun-Jong Lee · Jae-Han Lim

ICML 2026 · Seoul, South Korea

KWANGWOON UNIVERSITY

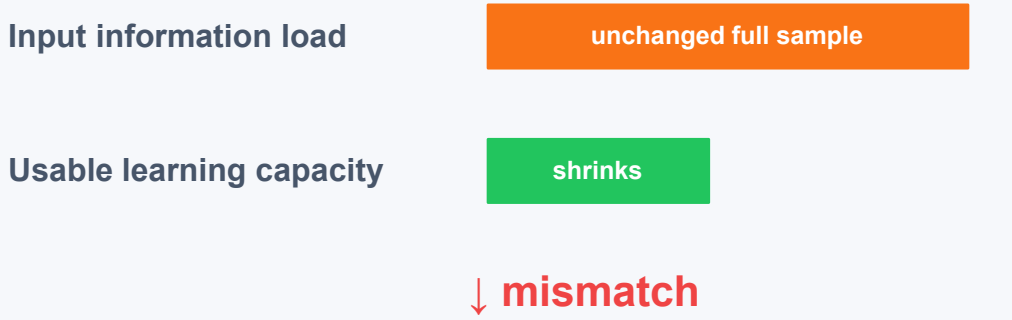
PROBLEM

Faults create an input-capacity mismatch

Practical gap

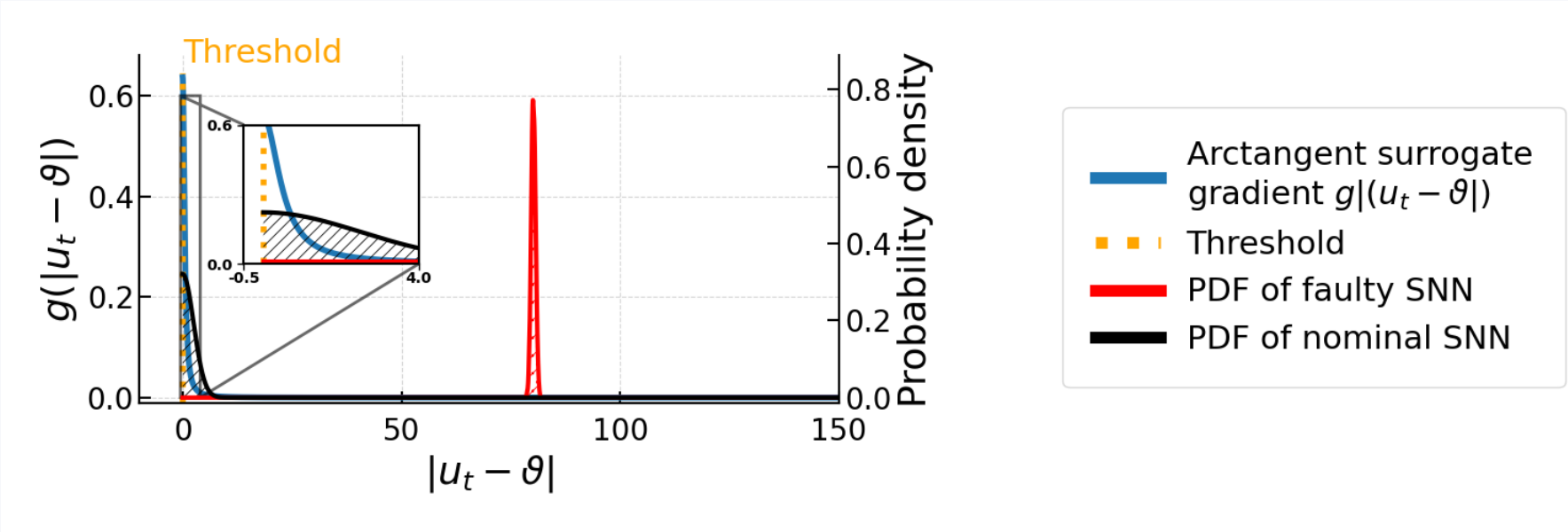
Many prior fixes require synapse rewriting,
neuron-state monitoring, rerouting, or recovery circuits.

Bottleneck



Attack the bottleneck before the circuit: regulate how the input is presented.

Evidence: faults push learning outside the gradient corridor



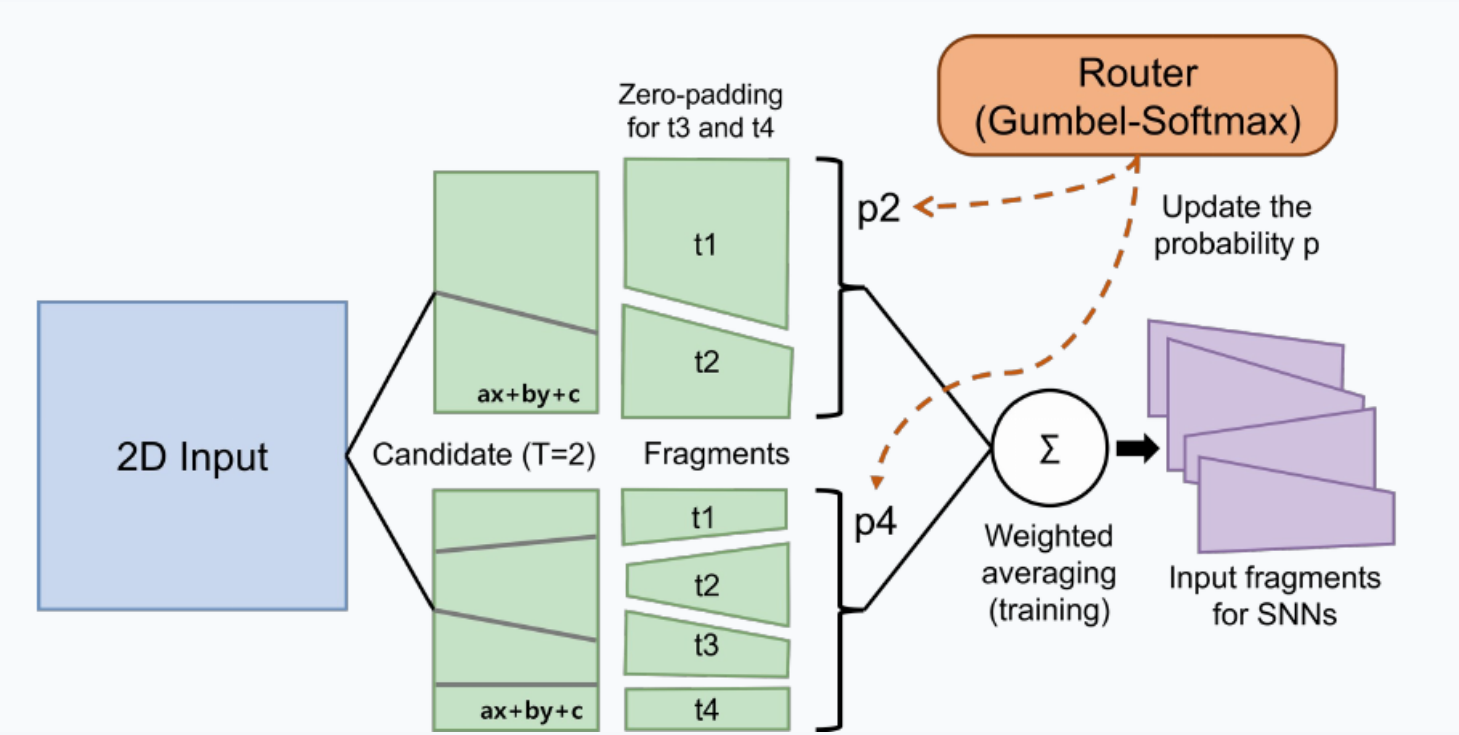
Interpretation

Faults do not merely corrupt weights; they move training into a near-zero-gradient regime.
Fragmenting the input reduces the instantaneous current and brings potentials back toward the threshold corridor.

Learn the input-side flow control strategy

What is learned?

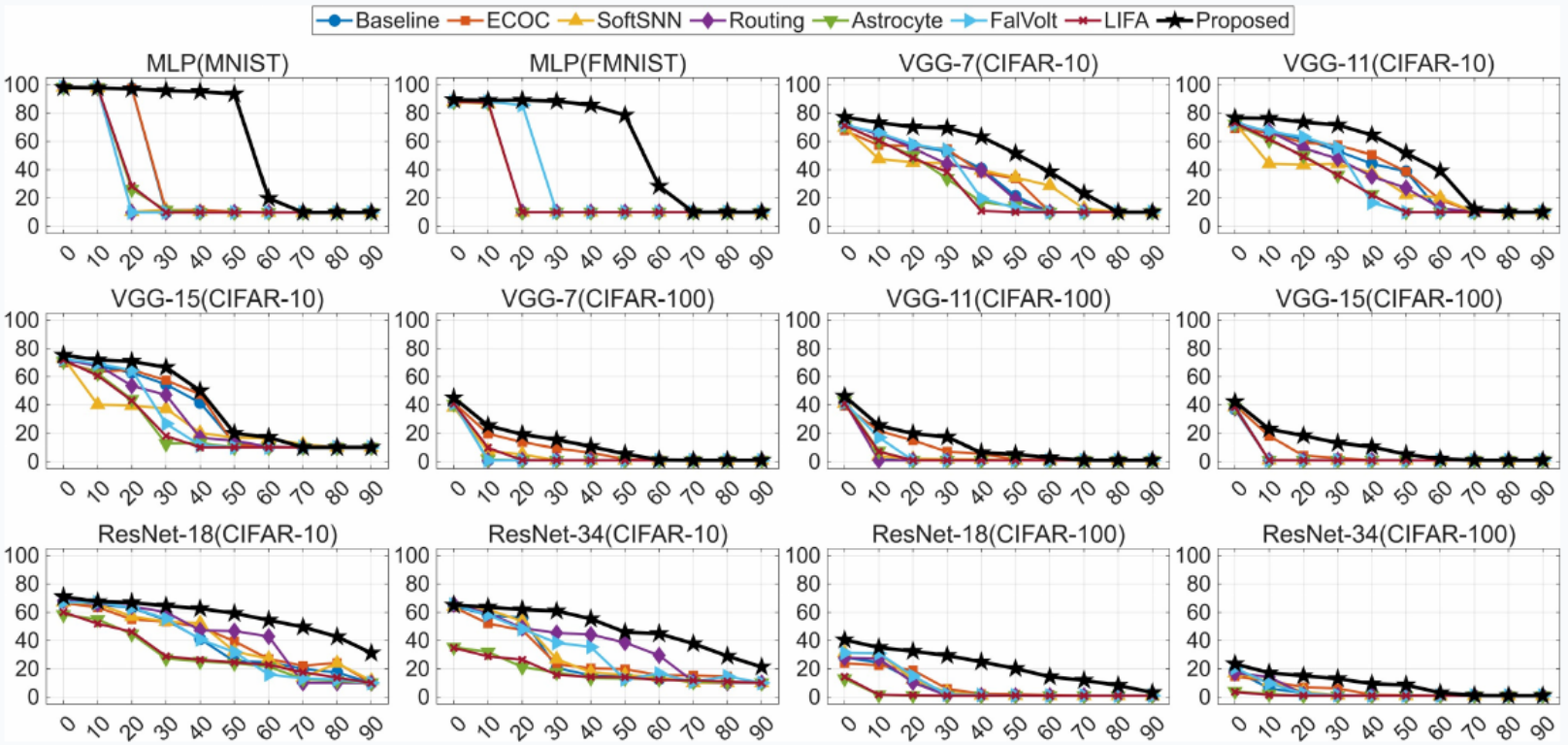
1. **Division lines**
   where to split the input sample
2. **Fragment count T**
   how many temporal pieces to use
3. **Balance loss**
   avoid dumping energy into one fragment
4. **Entropy decoding**
   weight confident time-step outputs



Training explores T ∈ {2, 4, 8} with Gumbel-Softmax; inference commits to one T and learned boundaries.

No synapse rewrite · No neuron-state scan · No routing map

Accuracy improves across models and fault ratios



Main SAF result

Across 12 panels, the black proposed curve stays above the baseline and benchmarks as the fault ratio increases.

21.4 → 51.5%

VGG-7 / CIFAR-10 at 50% SAF

Result is not a single-table gain; it is a consistent shift of the accuracy–fault curve.

Practicality and takeaway

- ✓ Multiple models and datasets
- ✓ SAF, RWF, and CEF fault types
- ✓ Real FPGA and Vivado emulator
- ✓ Each component contributes to ablation

No internal SNN-circuit access is required.

Control the information load — not the faulty circuit.

Thank you.