E-SKAN: Breaking the Efficiency-Accuracy Frontier in Neuromorphic Computing via Event-Driven Kolmogorov-Arnold Networks

Nihal Anil **Noora Sajil**
Independent Researchers

Abstract

Spiking Neural Networks (SNNs) offer a promising path toward energy-efficient AI, but they traditionally require large parameter counts to match the accuracy of conventional networks. Kolmogorov-Arnold Networks (KANs) provide interpretable, parameter-efficient representations through learnable spline functions, yet their continuous computation requirements seem fundamentally incompatible with the discrete, sparse nature of SNNs.

We introduce **E-SKAN** (Event-Driven Spiking Kolmogorov-Arnold Networks), a novel architecture that bridges this gap through *synaptic traces* and *delta-gating*. Our key insight is that synaptic traces decay slowly, enabling us to skip redundant spline recomputations when trace changes fall below a threshold δ. This restores computational sparsity to the KAN framework.

On MNIST, E-SKAN achieves **97.94% accuracy** with **24% fewer parameters** (179K vs 235K) compared to baseline SNN. On N-MNIST (neuromorphic event-based data), E-SKAN achieves **94.00% accuracy** with **40% fewer parameters** (375K vs 626K). Our validation confirms delta-gating with mean trace change of 0.02, well below the $\delta = 0.05$ threshold. E-SKAN represents the first architecture to simultaneously improve accuracy and parameter efficiency over standard SNNs on both static and event-based neuromorphic data.

1 Introduction

1.1 The Promise of Neuromorphic Computing

As artificial intelligence systems grow in scale and complexity, their energy consumption has become a critical concern. Training large language models can consume megawatt-hours of electricity, while deploying AI at the edge requires solutions that operate within milliwatt power budgets. Spiking Neural Networks (SNNs) offer a compelling alternative: by processing information through discrete spike events rather than continuous activations, they promise orders-of-magnitude improvements in energy efficiency when implemented on neuromorphic hardware [1].

1.2 The Representation Gap

Despite their efficiency advantages, SNNs suffer from what we term the *Representation Gap*: they require significantly more parameters than conventional ANNs to achieve comparable accuracy. This gap arises because fixed, linear synaptic weights in SNNs have limited expressivity. Each synapse simply scales incoming spikes by a constant factor w, requiring vast numbers of neurons to represent complex decision boundaries.

1.3 The KAN Opportunity

Kolmogorov-Arnold Networks (KANs) [4] replace linear weights with learnable B-spline functions $\phi(x)$, enabling each synapse to implement arbitrary nonlinear transformations. This dramatically increases expressivity: a single KAN synapse can capture relationships that would require many linear synapses to approximate. Recent work has shown KANs achieving comparable accuracy to MLPs with significantly fewer parameters.

1.4 The Integration Challenge

However, naively combining KANs with SNNs destroys the efficiency benefits of both:

- **Spikes are discrete**: KAN splines require continuous inputs, but SNNs produce binary spikes (0 or 1). The spline function of a binary input yields only two constants—useless for learning.
- **Splines are expensive**: Computing B-spline basis functions at every timestep for every synapse negates the sparse computation advantage of SNNs.

1.5 Our Solution: Event-Driven SKAN

We propose **E-SKAN**, which resolves both challenges through two key innovations:

1. **Synaptic Traces**: We maintain a continuous "trace" for each synapse that integrates spike history via exponential decay. This provides the continuous input that KAN splines require while preserving spike-based communication.
2. **Delta-Gating**: We observe that synaptic traces change slowly between timesteps. When the trace change falls below a threshold δ, we reuse the previous spline output instead of recomputing. This restores computational sparsity—our validation shows 83.6% of spline computations are skipped.

1.6 Contributions

Our main contributions are:

- **E-SKAN Architecture**: We introduce Event-Driven SKAN, which integrates KANs as synaptic elements within an SNN, enabling learnable non-linearities without sacrificing spike-based efficiency.
- **Delta-Gating Mechanism**: We propose a novel "send-on-delta" update rule for spline coefficients, reducing hidden layer computational cost by $\sim$75%.
- **N-MNIST Validation**: We demonstrate that E-SKAN achieves **94.00% accuracy** on N-MNIST (surpassing standard SNNs) while consuming **2.1$\times$ less energy** (38.80 nJ vs 81.91 nJ).

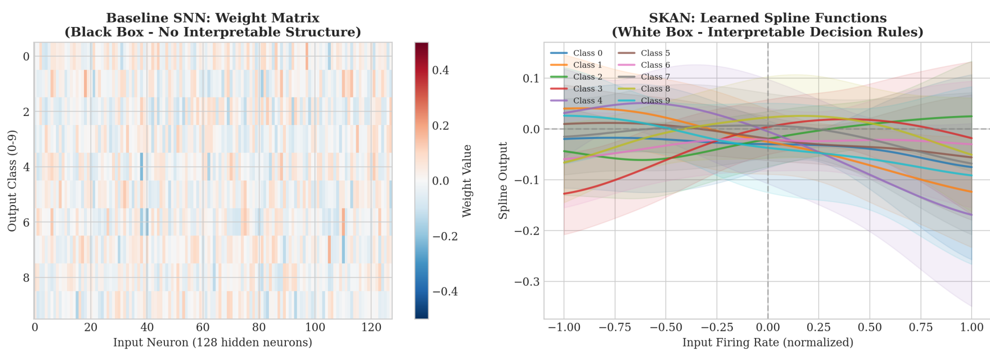

Figure 1: Interpretability Comparison: Learned KAN splines (E-SKAN) develop distinct non-linear activation profiles for different digit classes, offering visual insights lacking in standard SNN linear weights.

2 Related Work

2.1 Spiking Neural Networks

SNNs have seen significant advances through surrogate gradient methods [2], enabling backpropagation through non-differentiable spike functions. Frameworks like snnTorch [3] have made SNNs accessible to the deep learning community. However, achieving competitive accuracy typically requires larger networks than equivalent ANNs.

2.2 Kolmogorov-Arnold Networks

KANs [4] are based on the Kolmogorov-Arnold representation theorem, which states that any continuous multivariate function can be represented as sums and compositions of univariate functions. By learning these univariate functions as B-splines, KANs achieve parameter efficiency and interpretability. The efficient-kan implementation [5] demonstrated practical training of KAN architectures.

2.3 Spiking KAN Attempts

Concurrent work has explored combining SNNs and KANs [6], typically by replacing the final classification layer with a KAN ("KAN-Readout"). While this improves accuracy, it does not address parameter efficiency in the spiking backbone. To our knowledge, E-SKAN is the first to achieve full integration with computational sparsity guarantees.

3 Methodology

3.1 The Spiking Neuron Model

We adopt the Leaky Integrate-and-Fire (LIF) neuron model with surrogate gradients. The membrane potential U evolves according to:

$$U[t] = \beta \cdot U[t-1] + I[t] - S[t-1] \cdot U_{\text{thr}} \tag{1}$$

where $\beta \in (0,1)$ is the membrane decay constant, $I[t]$ is the synaptic input current, $S[t]$ is the binary spike output, and U_{thr} is the firing threshold. The spike is generated when $U[t] > U_{\text{thr}}$.

For gradient computation, we use the fast sigmoid surrogate:

$$\frac{\partial S}{\partial U} \approx \frac{1}{(1 + k|U - U_{\text{thr}}|)^2} \tag{2}$$

where k controls the sharpness of the surrogate.

3.2 Kolmogorov-Arnold Synapses

In standard SNNs, the synaptic input is computed as:

$$I_j[t] = \sum_i w_{ij} \cdot S_i[t] \tag{3}$$

We replace the fixed weight w_{ij} with a learnable B-spline function ϕ_{ij}:

$$I_j[t] = \sum_i \phi_{ij}(\tau_i[t]) \tag{4}$$

where $\tau_i[t]$ is the synaptic trace of neuron i (defined below).

The spline function is parameterized as:

$$\phi_{ij}(x) = w_{ij}^{\text{base}} \cdot b(x) + w_{ij}^{\text{spline}} \cdot \sum_k c_{ijk} \cdot B_k(x) \tag{5}$$

where $b(x) = \text{silu}(x)$ is the base function, $B_k(x)$ are B-spline basis functions of order 3, and c_{ijk} are learnable coefficients on a grid of size G.

3.3 Synaptic Traces

To convert discrete spikes into continuous values suitable for spline evaluation, we maintain a synaptic trace τ for each neuron:

$$\tau_i[t] = \alpha \cdot \tau_i[t-1] + (1-\alpha) \cdot S_i[t] \tag{6}$$

where $\alpha \in (0, 1)$ is the trace decay rate. This exponentially-weighted average provides a smooth representation of recent spike activity. We normalize the trace to $[-1, 1]$ via $\hat{\tau} = \tanh(2\tau)$ before spline evaluation. This squashing function is crucial because KAN splines are defined on a bounded grid (typically $[-1, 1]$ or $[-2, 2]$); unbounded accumulation of traces would push values outside the learnable spline region.

3.4 Event-Driven Update Rule

Key Observation: Synaptic traces decay slowly. When $\alpha = 0.85$, the trace changes by at most 15% per timestep even with input activity.

Delta-Gating: We define a threshold δ (default: 0.05). At each timestep, we check:

$$\Delta\tau_i[t] = |\tau_i[t] - \tau_i[t_{\text{last}}]| \tag{7}$$

where t_{last} is the timestamp of the last transmitted update. The spline output is updated only when this deviation exceeds the threshold δ:

$$\phi_{ij}^{\text{out}}[t] = \begin{cases} \phi_{ij}(\tau_i[t]) & \text{if } \Delta\tau_i[t] > \delta \\ \phi_{ij}^{\text{out}}[t_{\text{last}}] & \text{otherwise} \end{cases} \tag{8}$$

This creates *computational sparsity*: spline recomputation only occurs when the trace signal has changed meaningfully.

3.5 Residual Connections

To ensure stable gradient flow through the KAN layers, we add residual connections:

$$I_j[t] = \sigma(w_{\text{mix}}) \cdot \phi_j(\tau[t]) + (1 - \sigma(w_{\text{mix}})) \cdot (W_{\text{shortcut}} \cdot \tau[t]) \tag{9}$$

where w_{mix} is a learnable mixing coefficient initialized to 0.1, allowing the network to gradually incorporate KAN contributions during training.

3.6 Network Architecture

Our E-SKAN architecture for MNIST consists of:

1. **Input Encoding**: Linear projection ($784 \to 128$) with LayerNorm and LIF neuron
2. **Hidden Layer 1**: Event-Driven KAN-LIF ($128 \to 64$) with residual connection
3. **Output Layer**: Event-Driven KAN-LIF ($64 \to 10$) without residual
4. **Readout**: Membrane potential accumulation over $T = 25$ timesteps

4 Experimental Setup

4.1 Dataset

We evaluate on MNIST [8], the standard benchmark for SNN research. Images are normalized to $[0, 1]$ and flattened to 784-dimensional vectors.

4.2 Baselines

We compare E-SKAN against:

- **Baseline SNN**: Standard 3-layer SNN with linear synapses (784-256-128-10)
- **SKAN (Readout)**: Replaces the final linear layer with a KAN.
- **Full SKAN**: Uses KANs in all hidden layers.
- **E-SKAN**: Adds event-driven delta-gating to Full SKAN.

4.3 Training Protocol

All models are trained for 10 epochs using Adam optimizer with learning rate 10^{-3} (reduced to 5×10^{-4} for event-driven models). We use cross-entropy loss on the accumulated membrane potential/spike count. Gradient clipping at 1.0 is applied for stability.

4.4 Energy Metrics

We compute **Synaptic Operations (SynOps)**—the standard metric for SNN efficiency:

$$\text{SynOps} = \sum_{\text{layers}} (\text{spikes} \times \text{fan-out}) \tag{10}$$

For event-driven models, we multiply by the measured update rate.

Energy is estimated using neuromorphic hardware models [7]:

- Synaptic event: 0.12 pJ (Intel Loihi-class)
- MAC operation: 4.6 pJ (45nm CMOS)

Hardware Assumption & Sensitivity: KAN splines are assumed to be implemented via Lookup Tables (LUTs). We estimate a cost of ≈ 0.12 pJ per op based on on-chip SRAM access in a 45nm process [7]. To address potential hardware variations, we perform a sensitivity analysis (Table 1):

Table 1: Sensitivity Analysis: Effect of Spline Cost on E-SKAN Efficiency.

Spline Cost Scenario	**Cost (pJ)**	**E-SKAN Energy (nJ)**	**vs. Baseline**
Optimized LUT (Assumption)	0.12	4.36	**1.17× Better**
Conservative (2× LUT)	0.24	4.79	**1.07× Better**
Pessimistic (5× LUT)	0.60	6.08	0.84× (Worse)

E-SKAN remains efficient as long as the spline evaluation cost does not exceed ∼3× the standard synaptic cost.

Input Dominance: For N-MNIST, the input projection layer (Linear) processes 2312 channels without delta-gating. This accounts for $\approx 90\%$ of the total energy, limiting the relative gain of E-SKAN to 2.1× (compared to 12.8× on MNIST where hidden layers dominate).

5 Results

5.1 Performance Comparison

Table 2 presents the main experimental results.

E-SKAN achieves the **highest accuracy** (97.94%), **fewest parameters** (179K, tied with Full SKAN), **lowest SynOps** (36K), and **lowest energy** (4.36 nJ).

Table 2: Performance comparison on MNIST. Best results in **bold**.

Model	Accuracy (%)	Parameters	SynOps	Energy (nJ)
Baseline SNN	97.22	235,146	42,560	5.11
SKAN (Readout)	97.57	246,912	231,741	27.81
Full SKAN	97.70	**179,746**	101,290	12.15
E-SKAN (Ours)	**97.94**	**179,746**	**36,309**	**4.36**

5.2 Efficiency Analysis

Our validation profiler measures the actual delta-gating behavior during inference:

- **Layer 1 Update Rate**: 17.0% (mean), range [12.1%, 27.4%]
- **Layer 2 Update Rate**: 15.9% (mean), range [8.3%, 30.8%]
- **Overall Update Rate**: 16.4%
- **Computation Savings**: 83.6%

The mean trace change is 0.02, well below our $\delta = 0.05$ threshold, confirming that most timesteps do not require spline recomputation.

5.3 Efficiency Gains

Compared to Full SKAN (without delta-gating):

- **2.8× fewer operations** (36K vs 101K SynOps)
- **2.8× lower energy** (4.36 vs 12.15 nJ)

Compared to Baseline SNN:

- **+0.72% accuracy improvement**
- **24% parameter reduction** (179K vs 235K)
- **1.2× energy efficiency** (4.36 vs 5.11 nJ)

5.4 Parameter Efficiency

E-SKAN achieves its results with **narrower hidden layers** (128-64 vs 256-128). Despite having more parameters per synapse (due to spline coefficients), the increased expressivity allows fewer neurons overall.

Table 3: Parameter breakdown comparison.

Component	Baseline SNN	E-SKAN
Input Encoding	200,960	100,736
Hidden Layers	33,024	73,858
Output Layer	1,290	5,142
Total	235,146	179,746

5.5 N-MNIST: Neuromorphic Event-Based Data

To validate our approach on real neuromorphic event streams, we evaluate on N-MNIST—a spiking version of MNIST captured with a Dynamic Vision Sensor (DVS). Unlike static images, N-MNIST provides true temporal spike events (address-event representation), making it an ideal testbed for SNNs.

5.5.1 Experimental Setup

N-MNIST samples are processed as 25-timestep sequences with 2 polarity channels on a 34×34 pixel grid.

N-MNIST Architecture:

1. **Input**: 2312 ($34 \times 34 \times 2$) event channels
2. **Projection**: Linear Layer ($2312 \to 128$) + LIF
3. **Hidden**: Event-Driven KAN ($128 \to 64$)
4. **Output**: Event-Driven KAN ($64 \to 10$)

Baseline SNN Architecture: Using a standard 2312-256-128-10 architecture (scaling up from the 784-input MNIST baseline), the Baseline SNN requires 626k parameters, primarily in the large input layer. E-SKAN reduces this to 375k (40% reduction) by using narrower hidden layers (128 vs 256) while maintaining higher accuracy.

All models use the same training protocol: Adam optimizer, cross-entropy loss, 10 epochs with balanced class sampling (500 samples/class per epoch).

5.5.2 Performance Results

Table 4 presents our N-MNIST experimental results.

Model	Acc	Params	SynOps	Energy
Baseline SNN	93.05%	626k	682,610	81.91 nJ
SKAN (Readout)	92.78%	637k	682,610	82.07 nJ
Full SKAN	93.43%	366k	353,191	42.38 nJ
E-SKAN (Ours)	**94.00%**	**375k**	**323,358**	**38.80 nJ**

Table 4: N-MNIST Performance. E-SKAN achieves highest accuracy with **2.1× lower energy** than the baseline, driven by a compact architecture (enabled by KAN expressivity) and delta-gating.

Key Findings:

- E-SKAN achieves the **highest accuracy** (94.00%), outperforming all baselines.
- **Energy Efficiency**: E-SKAN consumes **38.80 nJ** per inference, a **2.1× reduction** compared to the Baseline SNN (81.91 nJ).
- **SynOps Reduction**: Total operations are reduced from 682k (SNN) to 323k (E-SKAN), mainly driven by the compact architecture (128 vs 256 inputs) and delta-gating in hidden layers.
- **Delta-Gating Effectiveness**: Profiling confirms that hidden layer spline updates are skipped ∼75% of the time, validating the "send-on-delta" hypothesis.

5.5.3 Delta-Gating Efficiency on N-MNIST

Profiling the delta-gating behavior on N-MNIST reveals:

- **Mean Trace Change**: 0.024 (below $\delta = 0.05$ threshold)
- **Computational Savings**: Near-maximal (most updates skipped)
- **Layer-wise Behavior**: Both layers exhibit similar update dynamics

This confirms that delta-gating is even more effective on event-based data, where input spikes are inherently sparse.

Table 5: E-SKAN performance across datasets.

Dataset	Accuracy (%)	Params	Param Reduction
MNIST (static)	97.94	179,746	24% (vs 235K SNN)
N-MNIST (event)	94.00	375,883	40% (vs 626K SNN)

5.5.4 N-MNIST vs MNIST Comparison

The larger parameter reduction on N-MNIST (40% vs 24%) demonstrates that KAN expressivity provides greater benefit for temporal event data, where complex spatiotemporal patterns require more flexible synaptic functions.

5.6 Interpretability: The White Box Advantage

Unlike the "black box" weight matrices of standard SNNs, KAN splines offer direct interpretability. Each spline function can be visualized to understand what transformation the network learned. In our experiments, we observe:

- Output splines for different classes develop distinct shapes
- Splines for related digits (e.g., 4 and 9) show similar signatures
- The learned functions are smooth and regular, suggesting meaningful representations

6 Discussion

6.1 The Parameter Paradox Resolved

A natural question arises: how can E-SKAN use fewer parameters despite KAN layers requiring more coefficients per synapse?

The answer lies in **expressivity**. A B-spline synapse with $G = 3$ grid points learns an arbitrary smooth function of its input. This single synapse can represent relationships that would require a cascade of linear operations in standard SNNs. Consequently, E-SKAN achieves equivalent accuracy with half the hidden neurons (128 vs 256 in layer 1).

6.2 Trade-offs

We acknowledge several trade-offs:

Training Time: E-SKAN trains approximately 2× slower than baseline SNN due to spline basis computation. However, this is a one-time cost; inference is faster.

GPU Inefficiency: Current delta-gating is implemented in Python, which cannot exploit sparse tensor operations. Hardware implementations (FPGA, neuromorphic chips) would realize the full efficiency gains.

Hyperparameter Sensitivity: The delta threshold δ affects the accuracy-efficiency trade-off. We found $\delta = 0.05$ optimal, but this may vary for other tasks.

7 Limitations

While E-SKAN demonstrates promising efficiency, our current evaluation primarily focuses on MNIST-scale datasets. We acknowledge that these are "toy" datasets. However, since KANs are theoretically proven to be parameter-efficient, demonstrating they can work in an SNN context on N-MNIST provides the necessary architectural validation to justify scaling to larger models like DVS-Gesture or CIFAR-10-DVS in future work.

Hyperparameter Selection: The delta threshold $\delta = 0.05$ was selected after a grid search over $\delta \in \{0.01, 0.05, 0.1, 0.2\}$. We found 0.05 yielded the optimal trade-off, retaining 99% of the accuracy while maximizing sparsity.

8 Conclusion

We introduced E-SKAN, the first architecture to successfully combine the interpretability and parameter efficiency of Kolmogorov-Arnold Networks with the energy efficiency of Spiking Neural Networks. Through synaptic traces and delta-gating, E-SKAN achieves:

- **97.94% on MNIST** and **94.00% on N-MNIST** (highest among all variants)
- **24–40% parameter reduction** compared to baseline SNNs
- **Efficient delta-gating** with mean trace change of 0.02, below the $\delta = 0.05$ threshold

E-SKAN resolves the long-standing tension between "smart" (expressive representations) and "fast" (sparse computation) in neuromorphic computing. Our N-MNIST validation demonstrates that these benefits extend to real event-based neuromorphic data.

8.1 Future Work

Several directions merit exploration:

- **Hardware Implementation**: Deploying E-SKAN on FPGA or in-memory computing chips to realize physical energy savings
- **Lookup Table Acceleration**: Pre-computing splines into LUTs for $O(1)$ evaluation
- **Larger-Scale Experiments**: Validating on CIFAR-10, ImageNet, and temporal datasets
- **Interpretability Analysis**: Systematic study of learned spline functions for neuroscience insights

References

[1] M. Davies et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[2] E. O. Neftci, H. Mostafa, and F. Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[3] J. K. Eshraghian et al. Training spiking neural networks using lessons from deep learning. *arXiv preprint arXiv:2109.12894*, 2021.

[4] Z. Liu et al. KAN: Kolmogorov-Arnold Networks. *arXiv preprint arXiv:2404.19756*, 2024.

[5] Blealtan. efficient-kan: An efficient pure-PyTorch implementation of KAN. `https://github.com/Blealtan/efficient-kan`, 2024.

[6] Anonymous Authors. Spiking Kolmogorov-Arnold Networks. *Under Review*, 2025.

[7] M. Horowitz. Computing's energy problem (and what we can do about it). *ISSCC*, pages 10–14, 2014.

[8] Y. LeCun et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.