SeMoE: Spike-aware Mixture-of-Experts for Energy-Efficient Speech Command Recognition

Anonymous ACL submission

Abstract

Always-on keyword recognition on battery-bounded edge devices is bottlenecked by inference energy, motivating spiking neural networks (SNNs) whose event-driven, accumulate-only operations replace dense multiplies with sparse spikes. Spike-driven transformers now approach published accuracy ceilings on spiking speech benchmarks, shifting the deployment bottleneck to per-sample energy, dominated by spike-driven attention that runs at full cost on every timestep, leaving speech's temporal redundancy unexploited. We introduce **SeMoE**, a spike-aware mixture-of-experts block that makes attention content-conditional along the temporal axis. A spiking gate, trained end-to-end with a straight-through estimator, routes each timestep to either a half-width sliding-window attention expert or an *identity expert* that bypasses attention; an expert-count sweep confirms this 2-expert design outperforms $K \in \{3, 4, 6\}$ alternatives. In a matched-pipeline ablation against a STASA attention baseline on SHD, SSC and GSC, SeMoE reduces inference energy by 30–49% and parameter count by 19–28%, achieves the lowest reported per-sample energy among comparable transformer-based spiking models, and remains within 0.76 pp of the strongest published accuracy on every benchmark, a gap narrower than the inter-method spread on these datasets. Content-conditional attention budgets complement firing-rate regularization and weight pruning, bringing always-on spoken-language understanding closer to edge deployment.

1 Introduction

Always-on speech command recognition on battery-bounded edge devices is bounded by inference energy, not accuracy (Garai and Samui, 2025). Spiking neural networks (SNNs) address this regime by replacing dense multiply–accumulates with sparse, accumulate-only spike operations (Hu et al., 2024), and recent spike-driven transformers (Zhou et al., 2022b; Wang et al., 2026) approach published accuracy ceilings on spiking speech benchmarks. Within these models, the spike-driven attention operator dominates the inference-energy budget: in multi-layer configurations, attention accounts for the majority of per-sample energy, making it the natural target for efficiency intervention.

Yet this attention is content-agnostic. Current spike-driven attention designs, exemplified by SpikCommander's spike-temporal-aware self-attention (STASA) (Wang et al., 2026), operate at full attention cost on every timestep, regardless of whether the frame carries phonetic structure or near-silence. Speech is bursty: information concentrates in transient events, so many timesteps are near-redundant. Conditional computation adapts effort to the input (Scardapane et al., 2024; Verelst and Tuytelaars, 2025), and mixture-of-experts is its dominant form (Cai et al., 2025). In the spiking domain, routing has been confined to spatial axes for vision (Zhou et al., 2024b), while efficiency work on spiking audio targets spike sparsification or weight pruning, leaving attention untouched. To our knowledge, no prior method makes spiking attention conditional on each timestep's content.

We close this gap with SeMoE,[1] a spike-aware mixture-of-experts block for SNN attention. A spiking gate, trained end-to-end with a straight-through estimator, dispatches each timestep to one of K experts using spike-activity statistics rather than learned dense embeddings. Its central mechanism is the *identity expert*: an expert that bypasses the attention operator and forwards the input through a single LIF activation, converting "do not attend" into a learnable per-timestep decision so that redundant frames incur no attention cost. Our minimal pool pairs the identity expert with a single half-width sliding-window expert.

[1] Code: https://anonymous.4open.science/r/scommand-A32A/.

Across three speech-command benchmarks (SHD, SSC (Cramer et al., 2022) and GSC (Warden, 2018)), a matched-pipeline ablation against a STASA attention baseline shows SeMoE reducing inference energy by 30–49% and parameter count by 19–28%, with an accompanying accuracy change of at most 0.73 pp (statistically significant on two of five configurations); SeMoE remains within 0.76 pp of the best published accuracy on each benchmark, a gap narrower than the spread separating methods on these datasets. It also attains the lowest reported per-sample energy among comparable transformer-based spiking models. Gate inspection confirms the mechanism: roughly two in five timesteps are routed to the identity expert. Together, these results position content-conditional attention as an orthogonal axis to firing-rate regularization and weight pruning for always-on spoken-language understanding on the edge. Our three technical contributions are:

1. **Spike-driven temporal routing for SNN attention.** A spiking gate driven by spike-activity statistics, rather than learned dense embeddings, dispatches each timestep to one of K experts, end-to-end spiking.

2. **The identity expert as a learnable attention-skip.** Including an identity expert in the pool turns "do not attend" into a per-timestep routing decision. Removing it costs accuracy even with more attention experts; identity acts as an essential regularizer, not a fallback.

3. **Small-pool saturation in our setup.** The $K{=}2$ pool already saturates routing capacity: $K{\in}\{3,4,6\}$ variants fail to recover its accuracy at higher training cost. This is regime-specific, not a universal MoE-scaling claim.

2 Related Work

2.1 Spiking Neural Networks for Speech Recognition

Spiking neural networks (SNNs) process information as sparse, asynchronous spikes, an energy-efficient alternative to conventional ANNs. Spikformer (Zhou et al., 2022b) brought the Transformer into this domain with a spike-form self-attention that removes the floating-point softmax, and later designs scaled spike-driven attention to competitive accuracy on vision benchmarks (Zhou et al., 2024a; Yao et al., 2024). For speech, Spik-Commander (Wang et al., 2026) adapts spike-driven attention to command recognition via a dual-branch spike-temporal-aware self-attention (STASA) module, reporting the highest published accuracy on the Spiking Heidelberg Digits (SHD), Spiking Speech Commands (SSC) (Cramer et al., 2022) and Google Speech Commands (GSC) (Warden, 2018) benchmarks; it is our baseline. Other spiking speech models target complementary axes: large-vocabulary recognition (Song et al., 2026), strong surrogate-gradient baselines (Bittar and Garner, 2022), and early-decision inference (Song et al., 2024). Recent 2025 work refines spike-driven attention further: spatio-temporal spike self-attention (ST-SSA) (Lee et al., 2025), addition-only attention (Guo et al., 2025), α-XNOR similarity (Xiao et al., 2025), and spike-timing token prioritization (Jang and Kim, 2025).

Across these designs, however, attention runs at every timestep: the operator pays a fixed cost regardless of whether a frame is phonetically salient. The sparsity of spike trains is exploited at the activation level but not at the routing level: these designs refine *how* attention computes, while the orthogonal question of *when* attention runs, decided from the input itself, remains open.

2.2 Mixture-of-Experts and Conditional Computation

Conditional computation routes each input through a subset of a model's parameters, decoupling capacity from per-input cost; mixture-of-experts (MoE) is its dominant form. The sparsely-gated MoE layer (Shazeer et al., 2017) learns a top-k gate over many experts; GShard (Lepikhin et al., 2020), the Switch Transformer (Fedus et al., 2022) and expert-choice routing (Zhou et al., 2022a) scale, simplify and rebalance this gating, and we adopt the Switch Transformer's load-balancing loss for our spike-driven gate. Related work varies per-input effort (Graves, 2017; Han et al., 2022) or pushes expert sparsity to individual neurons (Cheng et al., 2025). Closest to us, the spiking MoE of Zhou et al. (2024b) routes a spiking Transformer head- and channel-wise for vision; dedicated hardware and biologically-plausible gating for spiking MoE have also been studied (Xu et al., 2024; Bai et al., 2025), and Mixture-of-Depths (Raposo et al., 2024), in non-spiking Transformers, lets tokens skip layers. In non-spiking vision transformers, adaptive computation has also been realized through learnable

token halting (Yin et al., 2022) and similarity-based token merging (Bolya et al., 2023); both reduce cost by shrinking the token set rather than by gating computation within a fixed token grid.

SeMoE differs on two axes. First, the *routing axis*: prior spiking MoE routes along *spatial* axes (heads, channels), ST-SSA refines the attention pattern but still attends at every timestep (Lee et al., 2025), and Mixture-of-Depths (Raposo et al., 2024) selects, at each non-spiking Transformer layer, a top-k subset of tokens that pass through the layer's attention while the rest bypass it, trading computation across the *depth* dimension. SeMoE instead routes along the *temporal axis* of a single spike-driven attention block, dispatching each timestep to one of K experts that share the token grid but differ in compute cost.

Second, the *identity expert*: ours makes attention-skipping a learnable, per-timestep decision, whereas token-skip, token-merge, early-exit and adaptive-depth methods cut computation by reducing the effective token or layer count. SeMoE preserves both depth and the per-timestep token grid and routes around attention only when a timestep is uninformative, keeping the block fully spiking end-to-end.

2.3 Efficient Spiking Neural Networks

A large body of work reduces SNN inference cost. Energy is commonly estimated from synaptic operation counts using per-operation CMOS constants (Horowitz, 2014; Lemaire et al., 2023), a convention we follow while noting that such estimates can overstate gains on real hardware (Yan et al., 2026). Prior techniques act at the signal level, regularizing the firing rate to suppress redundant spikes (Yan et al., 2022), or at the weight level, pruning or quantizing spiking Transformers (Liu et al., 2023; Meng et al., 2023; Wei et al., 2025b; Castagnetti et al., 2025); hardware-aware sparsity (Wei et al., 2025a) and early-exit inference (Patne et al., 2026) are complementary directions.

SeMoE is complementary to all of these. It cuts cost at the *architectural* level by routing computation away from attention; because savings come from skipped operators rather than denser spikes or smaller weights, it stacks with firing-rate regularization or pruning.

3 Method

3.1 Preliminaries

A spiking neural network (SNN) unrolls its computation over T discrete timesteps and communicates through binary spike tensors. A leaky integrate-and-fire (LIF) neuron integrates a real-valued input into a membrane potential and emits a spike in $\{0, 1\}$ when the potential crosses threshold; we denote this activation $\mathrm{LIF}(\cdot)$ and train it with surrogate gradients (Neftci et al., 2019). Because activations are binary, spike-driven attention replaces the floating-point softmax and dense query–key product of standard attention with LIF gating and element-wise products, reducing a forward pass to sparse accumulate operations (Zhou et al., 2022b).

Our baseline is the STASA block of SpikCommander (Wang et al., 2026). From a spike input $\mathbf{x} \in \{0, 1\}^{T \times B \times D}$ with embedding dimension D, convolutional projections with batch normalization and LIF produce spiking queries, keys and values. STASA combines two parallel branches: a *sliding-window attention* (SWA) branch that, at each timestep t, sums queries and keys over a local window of radius w, scales the result, and gates the values through a LIF; and a *long-range attention* (LRA) branch that performs the same operation with sums taken over all T timesteps. The branch outputs are added and merged with a depthwise value convolution. This dual-branch operator runs in full at every timestep, applying the same compute budget to silent frames as to frames carrying salient phonemes.

3.2 The SeMoE Block

Spike activity is intrinsically non-uniform over time. Audio keywords interleave near-silence with short phonetic bursts, and downstream spike trains inherit this structure: many timesteps carry few or no spikes, while a small fraction concentrates most of the firing. SeMoE replaces STASA's uniform per-timestep operator with a pool of specialized experts and a lightweight spike-driven gate that routes each timestep through a single expert (Figure 1), realizing three design choices: **per-timestep routing**, so saved compute tracks the temporal granularity of spike activity; a **spike-driven gate** that consumes the spike tensor directly rather than a learned dense embedding as in standard mixture-of-experts (Shazeer et al., 2017; Fedus et al., 2022); and an **identity expert** that, when selected, converts spike sparsity into compute savings.

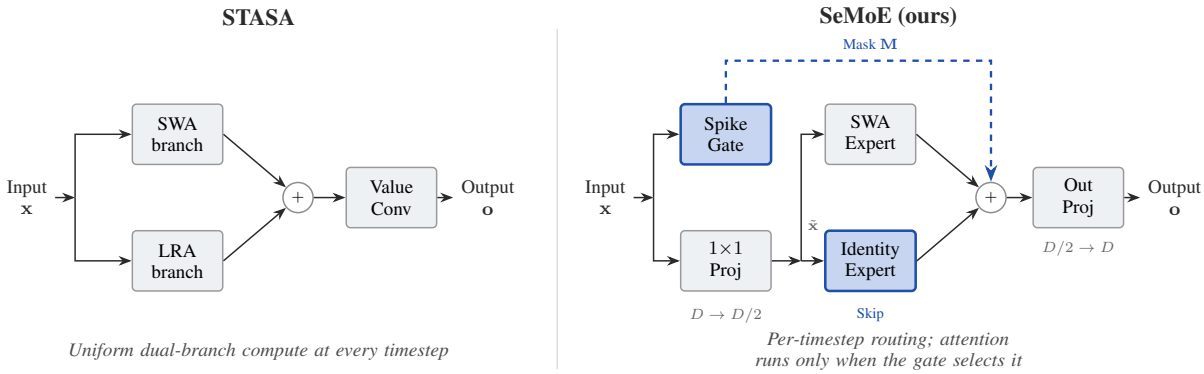

Figure 1: **Architecture.** A spike-driven gate (Figure 2) routes each timestep to a sliding-window attention (SWA) or identity (pass-through) expert. Blue marks SeMoE components.

Spike-driven gate. The gate decides, per timestep, which expert handles the frame (Figure 2). It is a lightweight spike-driven pipeline: a depthwise temporal convolution, batch normalization and LIF feed a linear map $\mathbf{W}_g \in \mathbb{R}^{K\times D}$ that produces the routing logits,

$$\boldsymbol{\ell}_t = \mathbf{W}_g\,\mathrm{LIF}\big(\mathrm{BN}(\mathrm{DWConv}(\mathbf{x}))\big)_t, \qquad \boldsymbol{\ell}_t \in \mathbb{R}^K. \tag{1}$$

Routing is top-1: timestep t is assigned to expert $\arg\max_k \ell_{t,k}$. Because $\arg\max$ has no useful gradient, we form the routing mask $\mathbf{M}$ with a straight-through estimator (Bengio et al., 2013), taking the gate's softmax as the differentiable surrogate,

$$\begin{aligned} \mathbf{M}_t &= \mathrm{onehot}\big(\arg\max_k \boldsymbol{\ell}_t\big) + \mathbf{s}_t - \mathrm{sg}(\mathbf{s}_t), \\ \mathbf{s}_t &= \mathrm{softmax}(\boldsymbol{\ell}_t). \end{aligned} \tag{2}$$

Here $\mathrm{sg}(\cdot)$ is the stop-gradient operator: the additive term $\mathbf{s}_t - \mathrm{sg}(\mathbf{s}_t)$ vanishes in the forward pass so $\mathbf{M}_t$ is the hard one-hot vector, while in the backward pass its gradient equals that of $\mathbf{s}_t$, so the gate routes discretely yet remains differentiable through the softmax. The gate's parameter and operation cost is a small fraction of the SWA expert's, and its full cost is included in the energy accounting.

Experts. A shared 1×1 convolution projects the input into a half-width space ($D_e = D/2$), denoted $\tilde{\mathbf{x}}$, that every expert reads; operating in D_e rather than D is the source of SeMoE's parameter reduction relative to STASA. We start with the smallest meaningful pool (one attention expert plus identity) to keep the gate's decision space minimal: $K{=}2$ experts comprising a *SWA expert* (the STASA sliding-window branch operating in D_e space with $w{=}20$) and an *identity expert* that passes the input through a single LIF with no attention parameters. Section 4.5 shows that enlarging this pool does not improve accuracy in our setup.

Output. The block output is the masked sum of the experts, projected back to dimension D,

$$\mathbf{o}_t = \mathrm{OutProj}\Big(\sum_{k=1}^{K} M_{t,k}\,[\mathrm{Expert}_k(\tilde{\mathbf{x}})]_t\Big). \tag{3}$$

A timestep routed to the identity expert bypasses the SWA computation entirely, converting the gate's choices into the per-timestep compute savings analyzed in Section 3.3.

Training. The network is trained end-to-end with a cross-entropy classification loss $\mathcal{L}_{\mathrm{CE}}$ and an auxiliary load-balancing loss that discourages the gate from collapsing. Following the Switch Transformer (Fedus et al., 2022), the term couples the mean soft gate probability $p_k = \mathbb{E}_t[\,\mathrm{softmax}(\boldsymbol{\ell}_t)_k\,]$ with the mean hard routing rate $f_k = \mathbb{E}_t[\,M_{t,k}\,]$, both averaged over timesteps and the batch,

$$\mathcal{L}_{\mathrm{LB}} = K\sum_{k=1}^{K} p_k\, f_k, \qquad \mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda\,\mathcal{L}_{\mathrm{LB}}, \tag{4}$$

with $\lambda = 0.01$. Gradient flows through LIF surrogate gradients and the STE of Eq. (2).

3.3 Energy Accounting

We adopt the standard inference-energy convention of the SNN literature (Lemaire et al., 2023; Horowitz, 2014). Energy is the synaptic-operation count weighted by per-operation constants,

$$E = E_{\mathrm{AC}} \cdot \mathrm{SOPs} + E_{\mathrm{MAC}} \cdot \mathrm{FLOPs}_{\mathrm{in}}, \tag{5}$$

where SOPs counts spike-driven accumulate operations across the network, $\mathrm{FLOPs}_{\mathrm{in}}$ counts

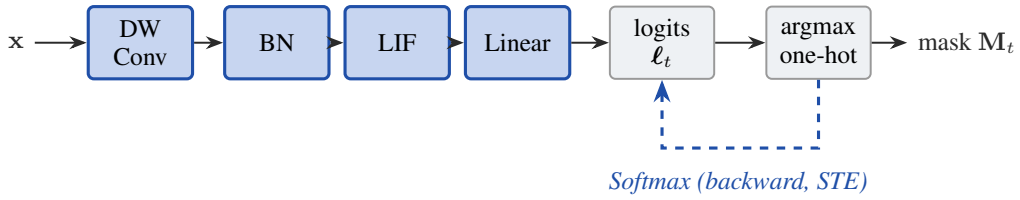

Figure 2: **Spike-driven gate.** Forward path yields the one-hot mask $\mathbf{M}_t$; dashed = STE backward via softmax$(\boldsymbol{\ell}_t)$.

the first-layer multiply–accumulates incurred on real-valued input, and E_{AC}, E_{MAC} are the per-operation 45 nm-CMOS energies of Horowitz (2014) (numerical values in Appendix B). Every spike-driven layer contributes accumulate (AC) operations weighted by its firing rate r: each output unit accumulates only the inputs that emit a spike, so the SOP count scales with r rather than with the layer's dense size, and silent inputs cost nothing. The first layer additionally contributes multiply–accumulate (MAC) operations when it consumes the real-valued Mel input (GSC); spike-train inputs (SHD, SSC) incur no MAC term.

SeMoE adds a single routing adjustment to this accounting. Operations inside expert k at layer ℓ are incurred only on the timesteps routed to it, so we weight them by the gate's mean usage u_k, giving an *effective firing rate* $r_{\mathrm{eff},k,\ell} = u_k\, r_\ell$. Non-routed components (the gate, the shared projections and the MLP) are counted in full at r_ℓ. Table 4 reports layer-averaged values; the per-layer breakdown and derivation are in Appendix B. We report energy as our primary efficiency metric because the deployment target is event-driven neuromorphic silicon, where GPU latency is not predictive of on-device cost (Lemaire et al., 2023).

4 Experiments

4.1 Setup

Datasets. We use Spiking Heidelberg Digits (SHD) and Spiking Speech Commands (SSC) (Cramer et al., 2022), two neuromorphic auditory datasets, and Google Speech Commands v2 (GSC) (Warden, 2018), a 35-class keyword dataset encoded as Mel features. Every input is processed over $T{=}100$ timesteps with 140 channels.

Training. All models are trained with AdamW and a cosine-annealing learning rate on a single RTX 5090 GPU (32 GB), with batch size 256, dropout 0.1, and per-dataset epochs and peak learning rate listed in Appendix A; test accuracies are reported as the mean over three seeds $\{0, 1, 2\}$.

Baseline and comparison protocol. Our baseline replaces the SeMoE block with the STASA attention block and holds every other component fixed, so any difference in accuracy, parameter count or inference energy is attributable to the routing mechanism. We report both encoder depths, with one or two stacked attention layers (1L, 2L).

4.2 Main Results

Accuracy across the field. On SHD, SeMoE reaches 96.04% at only 0.14M parameters, ranking top-3 across all reported models, behind only Pfa-SNN (96.26%) and SpikCommander 1L (96.41%), and matching the published 5-seed mean of 95.88± 0.39% (Wang et al., 2026) within seed variance. On SSC-2L (82.73%) and GSC-2L (96.44%) it stays within 0.76 pp of SpikCommander 2L (83.49% and 96.92%, respectively), at 1.70M parameters, a 20% reduction below that same two-layer transformer-based baseline (Table 1). Across depths, SeMoE preserves accuracy while spending fewer parameters, sitting in the strongest accuracy bracket at the smallest parameter footprint within its architectural family.

Dataset	Model	Param (M)	Time Steps	Acc (%)
SHD	SDT (1L) (Yao et al., 2023)	1.77	100	89.61
	Spikformer (1L) (Zhou et al., 2022b)	1.77	100	90.10
	DH-SNN (2L) (Zheng et al., 2024)	**0.05**	1000	92.10
	d-cAdLIF (2L) (Deckers et al., 2023)	0.08	100	94.85
	DCLS-Delays (2L) (Hammouamri et al., 2024)	0.20	100	95.07
	SpikeSCR (1L) (Wang et al., 2024)	0.26	100	95.60
	SE-adLIF (2L) (Baronig et al., 2025)	0.45	250	95.81
	Event-SSM^A (Schöne et al., 2024)	0.40	—	95.90
	Pfa-SNN (Sun et al., 2025)	0.20	100	96.26
	SpikCommander (1L) (Wang et al., 2026)	0.19	100	**96.41**
SSC	SDT (2L) (Yao et al., 2023)	2.57	100	79.82
	DCLS-Delays (2L) (Hammouamri et al., 2024)	1.40	100	80.16
	Spikformer (2L) (Zhou et al., 2022b)	2.57	100	80.18
	Pfa-SNN (Sun et al., 2025)	0.71	100	80.18
	d-cAdLIF (2L) (Deckers et al., 2023)	**0.70**	100	80.23
	SE-adLIF (2L) (Baronig et al., 2025)	1.60	250	80.44
	DCLS-Delays (3L) (Hammouamri et al., 2024)	2.50	100	80.69
	DH-SNN (3L) (Zheng et al., 2024)	0.35	1000	82.46
	SpikeSCR (1L) (Wang et al., 2024)	1.71	100	82.54
	SpikeSCR (2L) (Wang et al., 2024)	3.30	100	82.79
	SpikCommander (1L) (Wang et al., 2026)	1.12	100	83.26
	SpikCommander (2L) (Wang et al., 2026)	2.13	100	**83.49**
GSC	Spikformer (2L) (Zhou et al., 2022b)	2.57	100	91.86
	SDT (2L) (Yao et al., 2023)	2.57	100	91.88
	DH-SNN (3L) (Zheng et al., 2024)	**0.11**	1000	94.05
	DCLS-Delays (2L) (Hammouamri et al., 2024)	1.40	100	95.00
	DCLS-Delays (3L) (Hammouamri et al., 2024)	2.50	100	95.35
	SpikeSCR (1L) (Wang et al., 2024)	1.71	100	95.46
	SpikeSCR (2L) (Wang et al., 2024)	3.30	100	95.60
	d-cAdLIF (2L) (Deckers et al., 2023)	0.61	100	95.69
	Spiking LMUFormer (Liu et al., 2024)	1.69	—	96.12
	LMUFormer^A (Liu et al., 2024)	1.62	—	96.53
	SpikCommander (1L) (Wang et al., 2026)	1.12	100	96.71
	SpikCommander (2L) (Wang et al., 2026)	2.13	100	**96.92**

Table 1: Accuracy comparison on SHD, SSC and GSC.

Configuration	STASA Acc (%)	SeMoE Acc (%)	Δ Acc (pp, p)	Δ Params	Δ Energy
SHD-1L	95.45±0.55	96.04±0.41	+0.59 (0.17)	−28%	−49.5%
SSC-1L	83.20±0.19	82.66±0.29	−0.54 (0.17)	−19%	−29.8%
SSC-2L	83.46±0.15	82.73±0.19	−0.73 (0.004)	−20%	−47.5%
GSC-1L†	96.46±0.06	96.26±0.16	−0.20 (0.29)	−19%	−41.9%
GSC-2L	96.70±0.19	96.44±0.16	−0.26 (0.012)	−20%	−44.6%

Table 2: Same-pipeline ablation: SeMoE vs STASA. Bold p: α=0.05; † n=2 seeds.

Same-pipeline isolation. To attribute the gain to the routing mechanism rather than the surrounding encoder, we swap SeMoE for STASA in the same pipeline and compare under matched seeds (Table 2). The routing reduces inference energy by 30–49% and parameter count by 19–28% across every configuration, deterministic and uniform. Paired t-tests are reported with effect sizes; the accompanying accuracy change is at most 0.73 pp, and the two statistically significant gaps (0.26 pp on GSC-2L, 0.73 pp on SSC-2L) sit within the inter-method spread on these benchmarks (Table 1). The strongest accuracy–energy operating point is GSC-2L, where a 44.6% energy reduction sits within 0.48 pp of the highest published accuracy at a stable routing gate (Section 4.4).

4.3 Inference Energy in Context

On all three benchmarks SeMoE attains the lowest reported inference energy among comparable transformer-based SNN models at matching encoder depth (Table 3): 0.00161 mJ on SHD, 0.01359 mJ on SSC-2L and 0.01883 mJ on GSC-2L. That a routing-only change at the attention block delivers a 30–49% total-energy reduction (Section 4.2) confirms attention as the dominant per-sample energy contributor in these models. The reduction is computed within a single SOPs-accounting scope; cross-paper energy figures vary by codebase (e.g., whether the input projection or classification head is counted), so we report them as point estimates rather than as ratio claims.

Dataset	Model	Param (M)	SOPs (G)	Energy (mJ)
SHD	Spikformer (1L) (Zhou et al., 2022b)	1.77	0.2054	0.1849
	SDT (1L) (Yao et al., 2023)	1.77	0.1847	0.1663
	SpikeSCR (1L) (Wang et al., 2024)	0.26	0.0056	0.0051
	SpikCommander (1L) [published] (Wang et al., 2026)	0.19	0.0034	0.0039
	STASA (1L) [our meas.]	0.19	0.0036	0.00319
	SeMoE (1L) (ours)	**0.14**	**0.0018**	**0.00161**
SSC	Spikformer (2L) (Zhou et al., 2022b)	2.57	0.3169	0.2853
	SDT (2L) (Yao et al., 2023)	2.57	0.3084	0.2776
	SpikeSCR (2L) (Wang et al., 2024)	3.30	0.0348	0.0314
	SpikCommander (2L) [published] (Wang et al., 2026)	2.13	0.0173	0.0180
	STASA (2L) [our meas.]	2.13	0.0288	0.02591
	SeMoE (2L) (ours)	**1.70**	**0.0151**	**0.01359**
GSC	SpikeSCR (Wang et al., 2024)	3.15	0.049	0.094
	Spiking LMUFormer (Liu et al., 2024)	1.69	0.031	0.059
	SpikCommander (1L) [published] (Wang et al., 2026)	1.12	0.008	0.028
	SpikCommander (2L) [published] (Wang et al., 2026)	2.13	0.020	0.042
	STASA (2L) [our meas.]	2.13	0.0368	0.03396
	SeMoE (2L) (ours)	**1.70**	0.0200	**0.01883**

Table 3: Inference energy (mJ/sample). Shaded = SeMoE; bold = lowest per metric at matching depth.

4.4 Routing Behavior Drives the Efficiency

The efficiency reductions arise from learned, content-conditional routing rather than from degenerate gating: the gate neither collapses to a single expert nor follows a uniform skipping rate.

A stable balanced split. Tracking expert usage on GSC-2L over training, the gate immediately converges to a 59/41 split between the SWA expert and the identity expert and stays within four percentage points of that split for all 300 epochs (Figure 3). Roughly two in five timesteps are routed to the identity expert (and therefore skipped) over the lifetime of the model, providing the activity-side mechanism the efficiency claims of Section 4.3 require.

Two compounding sources of energy reduction. The parameter side comes from expert compression: setting the SWA expert width to $D_e = D/2$ halves its weight matrices, yielding the 19–28% parameter reduction across configurations (Table 4). The activity side comes from the identity expert: timesteps it receives incur no attention computation, so the effective firing rate falls by 12–33% and synaptic operations by 29–50%. On SSC-2L the two combine cleanly: a 20% parameter reduction and a 33% effective firing rate reduction together yield a 47.5% energy reduction, and four of the five configurations land in the 42–49% energy band.

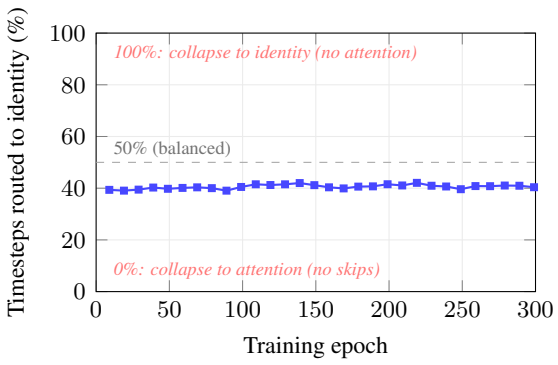

Figure 3: Identity-expert usage on GSC-2L (avg over 2 layers and 3 seeds). Stays near 40% throughout training: neither all-SWA (no skips) nor all-identity.

Routing adapts to available redundancy. SSC-1L sits at the low end of the band, and instructively so: with only a single attention layer, every attended frame carries the full phonetic burden, leaving little redundancy for the gate to skip. The gate responds adaptively, concentrating 87% of

Dataset	Depth	Method	Params (M)	Eff FR	SOPs (G)	Energy (mJ)	Δ E (%)
SHD	1L	STASA	0.191	0.1870	0.0036	0.00319	
SHD	1L	SeMoE	0.138	0.1360	0.0018	0.00161	−49.5
SSC	1L	STASA	1.120	0.1339	0.0150	0.01346	
SSC	1L	SeMoE	0.908	0.1172	0.0106	0.00945	−29.8
SSC	2L	STASA	2.127	0.1356	0.0288	0.02591	
SSC	2L	SeMoE	1.702	0.0914	0.0151	0.01359	−47.5
GSC	1L	STASA	1.120	0.1644	0.0185	0.01750	
GSC	1L	SeMoE	0.908	0.1171	0.0104	0.01016	−41.9
GSC	2L	STASA	2.127	0.1718	0.0368	0.03396	
GSC	2L	SeMoE	1.702	0.1207	0.0200	0.01883	−44.6

Table 4: Same-pipeline efficiency decomposition. Effective FR is the raw firing rate weighted by gate usage.

timesteps on the attention expert and routing only 13% through identity, and still delivers a 30% energy reduction together with the same 19% parameter saving. The same routing mechanism that finds two-in-five skippable timesteps in deeper encoders correctly finds far fewer in the shallow regime, so the energy gain tracks the redundancy available rather than being a uniform-rate skipping schedule.

4.5 Ablations and Robustness

Identity expert is essential. Removing the identity expert (the K3 variant with three attention experts) drops accuracy to 95.14% on SHD, below both K2 (96.04%) and STASA (95.45%), confirming that the value of routing lies in the ability to skip attention rather than in having multiple attention variants (Table 5).

More experts do not help. Increasing the expert count to K4 (95.21%) or K6 (95.67%) fails to recover K2's accuracy; widening experts to full dimension (95.56%) and removing the load-balancing loss (λ=0: 95.33%) also fail. K2 is therefore both the simplest and the strongest configuration in this sweep. Variant rows are single-seed point estimates used for qualitative ranking. Training cost stays within $\pm 25\%$ of the STASA baseline across configurations, while K4 roughly doubles wall-clock time (Appendix C).

Variant	Acc (%)	Configuration
K2 [SWA, identity]	96.04	main configuration
K3 [SWA, LRA, smallSWA] no-id	95.14	no-identity ablation
K4 [SWA, LRA, smallSWA, id]	95.21	default expert set
K4 fulld (no expert compression)	95.56	expert width ablation
K4 no load-balance (λ=0)	95.33	LB-weight ablation
K6 [SWA, LRA, smallSWA, id, +2]	95.67	extra experts
STASA baseline	95.45	non-routing baseline

Table 5: Expert-configuration ablation on SHD. K2 pairs one SWA expert with one identity expert. Multi-seed K2 numbers appear in Table 2.

Window-radius robustness. SeMoE inherits the sliding-window radius w from STASA; accuracy stays within a 0.9 pp band (95.42–96.28%) across $w \in \{4, \ldots, 50\}$ on SHD (Appendix D), confirming that the routing mechanism rather than a finely tuned window size drives the result.

Noise robustness. On GSC-2L under additive white Gaussian noise, SeMoE tracks the baseline within 0.3–1.2 pp across SNRs from clean to 0 dB (Appendix E); the energy reduction therefore carries no additional cost under realistic noise.

5 Conclusion

We introduced SeMoE, a spike-aware mixture-of-experts block that makes attention in spiking audio models conditional on input content. A spike-driven gate routes each timestep through a deliberately compact pool of specialized experts (in our configuration, a half-width sliding-window attention expert and an identity expert that skips attention entirely), so that computation follows the natural sparsity of the spike stream. Enlarging the expert count beyond this compact pool did not improve accuracy in our setup. Across SHD, SSC and GSC, SeMoE reduces inference energy by 30–49% and parameters by 19–28% in a matched-pipeline ablation against the STASA attention baseline, while remaining within 0.76 pp of the strongest published accuracy on each benchmark, a gap narrower than the inter-method spread on these datasets, with the two statistically significant configurations (0.26 and 0.73 pp) still falling inside that spread. SeMoE additionally attains the lowest reported per-sample inference energy among comparable transformer-based spiking models, and gate-usage analysis confirms that this reduction comes from learned, content-conditional skipping rather than from degenerate or uniform gating. The block preserves the input/output contract of standard spike-transformer attention and can therefore replace single-branch attention without changes to the surrounding encoder.

Because SeMoE reduces cost at the architectural level, it is complementary to signal-level methods such as firing-rate regularization, and combining the two is a natural next step. Extending the expert-pool ablation beyond SHD, evaluating transfer to other spike-driven attention designs, and deploying on neuromorphic hardware (replacing the silicon energy estimates with on-device measurements) would further test the approach.

Limitations

We highlight several limitations that scope the claims of this work.

Accuracy trade-off. On SSC and GSC, SeMoE's accuracy is 0.20–0.73 pp below the STASA baseline; the same-pipeline comparison in Section 4.2 reports per-configuration significance tests in Table 2. This effect size is consistent with our framing of SeMoE as an efficiency method (parameters and energy) rather than an accuracy method, and it sits inside the inter-method spread separating top published methods on these datasets (Table 1). The headroom for energy reduction is itself depth-dependent: in SSC-1L, a single attention layer must carry all phonetic structure, and the gate adaptively concentrates $\sim 87\%$ of timesteps on the attention expert (Section 4.4), still delivering a 30% energy reduction, but well below the 42–49% band achieved in deeper configurations. SeMoE's architectural benefit therefore scales with the redundancy available to the gate, and we position it for encoders with two or more attention layers.

Evaluation scope. We evaluate SeMoE as a substitute for one spike-driven attention operator (STASA); transfer to other spiking attention designs is unstudied. The evaluation is also confined to audio: SHD, SSC and GSC are speech-command benchmarks, and we do not test SeMoE on vision or event-camera (DVS) tasks. Expert-pool ablations are conducted on SHD as the standard spike-native benchmark for this comparison.

Measurement caveats. The energy figures are theoretical silicon estimates computed from synaptic-operation counts and CMOS constants (Section 3.3); on-device overheads on neuromorphic hardware such as Loihi or SpiNNaker may differ. Our seed budget is modest: results are averaged over three seeds, limited by single-GPU compute, so the significance tests of Section 4.2 have correspondingly low statistical power.

References

Qianyi Bai, Haiteng Wang, and Qiang Yu. 2025. A brain-inspired gating mechanism unlocks robust computation in spiking neural networks. *Preprint*, arXiv:2509.03281.

Maximilian Baronig, Romain Ferrand, Silvester Sabathiel, and Robert Legenstein. 2025. Advancing spatio-temporal processing in spiking neural networks through adaptation. *Nature Communications*, 16(1):5776.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *Preprint*, arXiv:1308.3432.

Alexandre Bittar and Philip N. Garner. 2022. A surrogate gradient spiking baseline for speech command recognition. *Frontiers in Neuroscience*, Volume 16 - 2022.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your ViT but faster. In *International Conference on Learning Representations (ICLR)*.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2025. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 37(7):3896–3915.

Andrea Castagnetti, Alain Pegatoquet, and Benoît Miramond. 2025. All in one timestep: Enhancing sparsity and energy efficiency in multi-level spiking neural networks. *Preprint*, arXiv:2510.24637.

Runxi Cheng, Yuchen Guan, Yucheng Ding, Qingguo Hu, Yongxian Wei, Chun Yuan, Yelong Shen, Weizhu Chen, and Yeyun Gong. 2025. Mixture of neuron experts. *Preprint*, arXiv:2510.05781.

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. 2022. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757.

Lucas Deckers, Laurens Van Damme, Ing Jyh Tsang, Werner Van Leekwijck, and Steven Latré. 2023. Co-learning synaptic delays, weights and adaptation in spiking neural networks. *Preprint*, arXiv:2311.16112.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1).

Soumen Garai and Suman Samui. 2025. Advances in small-footprint keyword spotting: A comprehensive review of efficient models and algorithms. *Preprint*, arXiv:2506.11169.

Alex Graves. 2017. Adaptive computation time for recurrent neural networks. *Preprint*, arXiv:1603.08983.

Yufei Guo, Xiaode Liu, Yuanpei Chen, Weihang Peng, Yuhan Zhang, and Zhe Ma. 2025. Spiking transformer: Introducing accurate addition-only spiking self-attention for transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24398–24408.

Ilyass Hammouamri, Ismail Khalfaoui-Hassani, and Timothée Masquelier. 2024. Learning delays in spiking neural networks using dilated convolutions with learnable spacings. In *International Conference on Learning Representations (ICLR)*.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. 2022. Dynamic Neural Networks: A Survey . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(11):7436–7456.

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

Yangfan Hu, Qian Zheng, Guoqi Li, Huajin Tang, and Gang Pan. 2024. Toward large-scale spiking neural networks: A comprehensive survey and future directions. *Preprint*, arXiv:2409.02111.

Minsuk Jang and Changick Kim. 2025. Sparta: Advancing sparse attention in spiking neural networks via spike-timing-based prioritization. *Preprint*, arXiv:2508.01646.

Donghyun Lee, Yuhang Li, Youngeun Kim, Shiting Xiao, and Priyadarshini Panda. 2025. Spiking transformer with spatial-temporal attention. *Preprint*, arXiv:2409.19764.

Edgar Lemaire, Loïc Cordone, Andrea Castagnetti, Pierre-Emmanuel Novac, Jonathan Courtois, and Benoît Miramond. 2023. *An Analytical Estimation of Spiking Neural Networks Energy Efficiency*, page 574–587. Springer International Publishing.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *Preprint*, arXiv:2006.16668.

Yue Liu, Shanlin Xiao, Bo Li, and Zhiyi Yu. 2023. Sparsespikformer: A co-design framework for token and weight pruning in spiking transformer. *Preprint*, arXiv:2311.08806.

Zeyu Liu, Gourav Datta, Anni Li, and Peter Anthony Beerel. 2024. LMUFormer: Low complexity yet powerful spiking model with Legendre Memory Units. In *International Conference on Learning Representations (ICLR)*.

L. W. Meng, G. C. Qiao, X. Y. Zhang, J. Bai, Y. Zuo, P. J. Zhou, Y. Liu, and S. G. Hu. 2023. An efficient pruning and fine-tuning method for deep spiking neural network. *Applied Intelligence*, 53(23):28910–28923.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. 2019. 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.

Parth Patne, Mahdi Taheri, Ali Mahani, Maksim Jenihhin, Reza Mahani, and Christian Herglotz. 2026. Sparq: Spiking early-exit neural networks for energy-efficient edge ai. *Preprint*, arXiv:2603.14380.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. 2024. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *Preprint*, arXiv:2404.02258.

Simone Scardapane, Alessandro Baiocchi, Alessio Devoto, Valerio Marsocci, Pasquale Minervini, and Jary Pomponi. 2024. Conditional computation in neural networks: Principles and research trends. *Intelligenza Artificiale*, 18(1):175–190.

Mark Schöne, Neeraj Mohan Sushma, Jingyue Zhuge, Christian Mayr, Anand Subramoney, and David Kappel. 2024. Scalable event-by-event processing of neuromorphic sensory signals with deep state-space models. *Preprint*, arXiv:2404.18508.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Zeyang Song, Qianhui Liu, Qu Yang, Yizhou Peng, and Haizhou Li. 2024. Ed-skws: Early-decision spiking neural networks for rapid,and energy-efficient keyword spotting. *Preprint*, arXiv:2406.12726.

Zeyang Song, Shimin Zhang, Yuhong Chou, Jibin Wu, and Haizhou Li. 2026. Iml-spikeformer: Input-aware multilevel spiking transformer for speech processing. *IEEE Transactions on Neural Networks and Learning Systems*, 37(3):1377–1389.

Pengfei Sun, Jibin Wu, Paul Devos, and Dick Botteldooren. 2025. Towards parameter-free attentional spiking neural networks. *Neural Netw.*, 185(C).

Thomas Verelst and Tinne Tuytelaars. 2025. A survey on dynamic neural networks: from computer vision to multi-modal sensor fusion. *Preprint*, arXiv:2501.07451.

Jiaqi Wang, Liutao Yu, Liwei Huang, Chenlin Zhou, Han Zhang, Zhenxi Song, Min Zhang, Zhengyu Ma, and Zhiguo Zhang. 2024. Efficient speech command recognition leveraging spiking neural network and curriculum learning-based knowledge distillation. *Preprint*, arXiv:2412.12858.

Jiaqi Wang, Liutao Yu, Xiongri Shen, Sihang Guo, Chenlin Zhou, Leilei Zhao, Yi Zhong, Zhiguo Zhang, and Zhengyu Ma. 2026. Spikcommander: A high-performance spiking transformer with multi-view learning for efficient speech command recognition. *Preprint*, arXiv:2511.07883.

Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *Preprint*, arXiv:1804.03209.

Chiyue Wei, Bowen Duan, Cong Guo, Jingyang Zhang, Qingyue Song, Hai "Helen" Li, and Yiran Chen. 2025a. Phi: Leveraging pattern-based hierarchical sparsity for high-efficiency spiking neural networks. *Preprint*, arXiv:2505.10909.

Wenjie Wei, Malu Zhang, Zijian Zhou, Ammar Belatreche, Yimeng Shan, Yu Liang, Honglin Cao, Jieyuan Zhang, and Yang Yang. 2025b. Qp-snn: Quantized and pruned spiking neural networks. *Preprint*, arXiv:2502.05905.

Yichen Xiao, Shuai Wang, Dehao Zhang, Wenjie Wei, Yimeng Shan, Xiaoli Liu, Yulin Jiang, and Malu Zhang. 2025. Rethinking spiking self-attention mechanism: Implementing α-XNOR similarity calculation in spiking transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5444–5454.

Boxun Xu, Junyoung Hwang, Pruek Vanna-iampikul, Yuxuan Yin, Sung Kyu Lim, and Peng Li. 2024. Towards 3d acceleration for low-power mixture-of-experts and multi-head attention spiking transformers. *Preprint*, arXiv:2412.05540.

Yulong Yan, Haoming Chu, Yi Jin, Yuxiang Huan, Zhuo Zou, and Lirong Zheng. 2022. Backpropagation with sparsity regularization for spiking neural network learning. *Frontiers in Neuroscience*, Volume 16 - 2022.

Zhanglu Yan, Zhenyu Bai, and Weng-Fai Wong. 2026. Reconsidering the energy efficiency of spiking neural networks. *Preprint*, arXiv:2409.08290.

Man Yao, Jiakui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo Xu, and Guoqi Li. 2024. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. *Preprint*, arXiv:2404.03663.

Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. 2023. Spike-driven transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. 2022. A-ViT: Adaptive tokens for efficient vision transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10809–10818.

Hanle Zheng, Zhong Zheng, Rui Hu, Bo Xiao, Yujie Wu, Fangwen Yu, Xue Liu, Guoqi Li, and Lei Deng. 2024. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15(1).

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y. Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. 2022a. Mixture-of-experts with expert choice routing. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Zhaokun Zhou, Kaiwei Che, Wei Fang, Keyu Tian, Yuesheng Zhu, Shuicheng Yan, Yonghong Tian, and Li Yuan. 2024a. Spikformer v2: Join the high accuracy club on imagenet with an snn ticket. *Preprint*, arXiv:2401.02020.

Zhaokun Zhou, Yijie Lu, Yanhao Jia, Kaiwei Che, Jun Niu, Liwei Huang, Xinyu Shi, Yuesheng Zhu, Guoqi Li, Zhaofei Yu, and Li Yuan. 2024b. Spiking transformer with experts mixture. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. 2022b. Spikformer: When spiking neural network meets transformer. *Preprint*, arXiv:2209.15425.

A Hyperparameters and Reproducibility

Datasets differ mainly in learning rate, model dimension and encoder depth (Table 6). All configurations share the remaining hyperparameters: the AdamW optimizer under a cosine-annealing learning-rate schedule, batch size 256, dropout 0.1, and computation unrolled over $T = 100$ timesteps on 140-bin inputs. The leaky integrate-and-fire neuron uses membrane time constant $\tau = 2.0$, firing threshold 1.0, reset 0.5, and an arctangent surrogate gradient. By default every expert operates in the half-width space ($D_e = D/2$), and the sliding-window attention expert uses window radius $w = 20$.

	SHD	SSC	GSC
Epochs	500	300	300
Peak learning rate	1×10^{-2}	5×10^{-3}	2×10^{-3}
Weight decay	1×10^{-2}	1×10^{-2}	5×10^{-3}
Encoder depth	1	1, 2	1, 2
Model dimension D	128	256	256
Attention heads	8	16	16
MLP expansion	1	4	4

Table 6: Training and model hyperparameters.

The spike-driven gate is a depthwise temporal convolution (kernel size 3) followed by batch normalization, a LIF activation, and a linear map to K routing logits; routing is top-1 with a straight-through estimator. The load-balancing weight is $\lambda = 0.01$. Gradient clipping at norm 1.0 is applied to all SeMoE variants, which stabilizes the gate during the first epochs of training. Inputs are augmented with SpecAugment on the Mel features (GSC) and EventDrop on the spike trains (SHD and SSC).

Every reported accuracy is the mean over three seeds, with the standard deviation listed in Table 2. All runs use a single RTX 5090 GPU. Inference energy and synaptic-operation counts follow the procedure in App. B, applied identically to SeMoE and the STASA baseline. Preprocessing follows the published protocol for each benchmark so that every comparison isolates the attention block. Code: https://anonymous.4open.science/r/scommand-A32A/.

B Energy Estimation Details

Inference energy follows the standard SNN-literature convention (Lemaire et al., 2023; Horowitz, 2014); the general formula is Eq. (5) (Section 3.3).

For each spike-driven layer ℓ with firing rate r_ℓ and accumulate-op count $\mathrm{FLOPs}_{\mathrm{AC},\ell}$ at unit rate,

$$E_\ell = E_{\mathrm{AC}} \cdot r_\ell \cdot \mathrm{FLOPs}_{\mathrm{AC},\ell}, \tag{6}$$

summed over all layers; only the first layer (GSC Mel input) adds $E_{\mathrm{MAC}} \cdot \mathrm{FLOPs}_{\mathrm{in}}$ instead. Constants are $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$ and $E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$ at 45 nm.

For layers inside a SeMoE expert k, the fraction of timesteps that execute them is the gate's mean usage u_k, giving an effective firing rate $r_{\mathrm{eff},k,\ell} = u_k\, r_\ell$ and

$$E_{k,\ell} = E_{\mathrm{AC}} \cdot u_k\, r_\ell \cdot \mathrm{FLOPs}_{\mathrm{AC},\ell}. \tag{7}$$

Non-routed components (the gate, the shared projection and the MLP) use r_ℓ directly ($u_k = 1$). The identity expert has negligible $\mathrm{FLOPs}_{\mathrm{AC},\ell}$, so timesteps routed to it incur almost no energy. Table 4 reports the layer-averaged "Eff FR" for 2L models.

C Ablation Details

Our expert-pool ablation on SHD (Section 4.5) compares six variants: K2 is [SWA, identity]; K3 (no identity) is [SWA, LRA, small-window SWA]; K4 is [SWA, LRA, small-window SWA, identity]; K6 extends K4 with two further attention experts. The small-window SWA expert uses radius $w = 5$. Two K4 controls isolate design choices: K4-fulld drops the half-width projection (every expert at full width D), and K4 ($\lambda = 0$) removes load balancing. None of these improves on K2 (Table 5).

K2 trains within $\pm 25\%$ of the STASA baseline and well below the K4 expert set (Table 7); the K2–K4 gap exceeds the shared-RTX 5090 wall-clock variance.

Dataset	Depth	Method	N	Wall-clock (h)	Range (h)	Δ vs STASA
SHD	1L	STASA	9	0.82	0.35–1.03	—
SHD	1L	SeMoE	17	0.79	0.63–1.62	−3.7%
SHD	1L	K4	9	1.56	1.02–1.99	+90.2%
SSC	1L	STASA	3	6.95	5.98–7.47	—
SSC	1L	SeMoE	3	6.21	6.01–6.34	−10.6%
SSC	1L	K4	3	6.98	5.89–8.12	+0.4%
SSC	2L	STASA	3	7.61	7.45–7.76	—
SSC	2L	SeMoE	3	6.32	5.63–7.00	−17.0%
SSC	2L	K4	3	10.97	10.67–11.26	+44.2%
GSC	1L	STASA	3	7.71	7.41–7.89	—
GSC	1L	SeMoE	2	9.59	9.57–9.61	+24.4%
GSC	2L	STASA	3	9.66	9.25–9.91	—
GSC	2L	SeMoE	3	9.19	6.17–10.84	−4.9%
GSC	2L	K4	3	13.26	12.69–14.15	+37.3%

Table 7: Training wall-clock time (mean, min–max).

D Window-Radius Sensitivity

The SWA expert has one hyperparameter, the window radius w. On SHD K2, accuracy stays within a 0.9 pp band (95.42–96.28%) across $w \in \{4, \ldots, 50\}$ (Table 8); the default w=20 sits near the top, so routing rather than window tuning drives the result.

Window radius w	Acc (%) mean ± std	N seeds
4	95.80 ± 0.13	2
8	95.47 ± 0.01	2
12	95.84 ± 0.62	2
16	95.42 ± 0.38	2
20 ← default	95.99 ± 0.57	2
32	**96.28** ± 0.04	2
50	95.60 ± 0.16	2

Table 8: Sliding-window radius sensitivity for SeMoE K2 on SHD (2 seeds per radius).

E Noise Robustness

We test noise robustness on GSC-2L; SHD and SSC are spike-domain and admit a different perturbation model deferred to future work. Under additive white Gaussian noise from clean down to 0 dB, SeMoE tracks the STASA baseline (Table 9): the accuracy gap stays at 0.3–0.5 pp from clean to 10 dB, widens to 1.2 pp at 5 dB, and narrows to 0.4 pp at 0 dB. SeMoE's standard deviation under low SNR is also smaller than STASA's (0.39 vs. 1.02 at 5 dB; 1.87 vs. 3.12 at 0 dB), indicating more stable behavior at high noise.

Method	SNR clean	SNR 20 dB	SNR 10 dB	SNR 5 dB	SNR 0 dB
STASA-2L	96.70 ± 0.19	96.67 ± 0.12	94.94 ± 0.20	87.80 ± 1.02	60.44 ± 3.12
SeMoE-2L	96.44 ± 0.16	96.35 ± 0.08	94.46 ± 0.16	86.61 ± 0.39	60.00 ± 1.87

Table 9: Noise robustness on GSC-2L (additive WGN). Mean±std, 3 seeds.