ATTACHMENT POINT, NOT ROUTER: A CEILING ON CONDITIONAL COMPUTATION IN SPIKING SPEECH MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Conditional computation is widely proposed as a route to energy-efficient spiking speech transformers: a gate decides, per frame, whether the attention operator runs, and savings of 30 to 49% are attributed to the gate. We show that the mechanism cannot produce them. The operations a query-side router can remove are bounded by a product of three factors, $\Delta C/C \leq u_{\mathrm{id}}\, s_{\mathrm{attn}}\, \rho$, and the third has been overlooked. Because a frame's key and value are read by its neighbours, and because the projections bracketing the operator run regardless of the routing decision, only $\rho = 22$ to 28% of the attention operator belongs to the router in streaming speech models. Across eight audio and nine vision configurations from published architectures the ceiling is below 2.9% for every streaming-capable speech model and below 7% for every windowed architecture; it survives removing the width bottleneck, tightens under a better window implementation, and within this family admits no feasible configuration above 5.3% at the temporal resolution these benchmarks use. Energy is not operations, and we keep the two apart: forcing a trained gate across the whole skip axis removes 7.1% of operations and 10.9% of measured energy, while accuracy falls 9.8 points. An ablation ladder that adds one change at a time locates the 49.5% we ourselves recorded for this mechanism in a static width reduction, with a further 8 to 17.5 points of it an artefact of the energy formula in common use. Finally we reach the higher ceiling the analysis predicts. Holding the routing rule, rate and training protocol fixed and varying only where the mask is applied, one random rule removes 2.7% of energy at the attention operator, 11.0% at the channel MLP and 25.9% at the whole block, each within a few points of the ceiling derived for it, at accuracy flat to within a point. The obstacle is not the router but where it is attached.

1 INTRODUCTION

Keyword spotting on battery-bounded devices is constrained by inference energy rather than by accuracy. Spiking neural networks address this by replacing dense multiply–accumulates with sparse, accumulate-only operations, and spiking transformers have brought this regime to competitive accuracy on the standard speech-command benchmarks. Within these models, a natural next step is *conditional computation*. Speech is bursty, information concentrates in transient events, and many frames appear near-redundant, so a gate that decides per frame whether the attention operator runs promises energy savings at little accuracy cost. Recent work in this direction reports per-sample energy reductions of 30 to 49% and attributes them to exactly this mechanism.

This paper asks a prior question that, to our knowledge, has not been asked in this literature: how much energy could such a gate save even if it were perfect? The answer is a short calculation, and it is discouraging. Work on spiking attention operators, on conditional computation, and on energy accounting bears on it from three directions, and §5 positions the paper against each once the bound is in hand. The computation removable by any router that decides whether a frame attends is bounded by a product of three factors,

$$\frac{\Delta C}{C} \leq \underbrace{u_{\mathrm{id}}}_{\text{fraction routed away}} \times \underbrace{s_{\mathrm{attn}}}_{\text{attention's share of cost}} \times \underbrace{\rho}_{\text{share of attention the router owns}},$$

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and the third factor has gone unexamined. It is not close to one. A frame's key and value are read by every neighbouring query, so they must be computed whatever the router decides, and the projections bracketing the attention operator run per frame regardless, because the skipped path's output still has to re-enter the residual stream. In the streaming-capable speech models we examine, these two mechanisms leave the router in possession of only $\rho = 22$ to 28% of the operator it gates.

The ceiling is low and hard to escape. Evaluating the bound across eight audio and nine vision configurations drawn from published spiking transformers, under one formula and with every ambiguity resolved in attention's favour, the ceiling at a realistic 40% routing rate lies below 2.9% for every streaming-capable speech model and below 7% for every windowed architecture (§2.2). Three properties make it hard to escape (§2.3). It *survives removing the width bottleneck* that the method under study introduces, since taking experts back to full width nearly doubles ρ and still leaves the ceiling at 3.7 to 6.8%. It *tightens under a better implementation*, since the window accumulation is the only query-side term that grows with the window, so replacing a materialised window with a prefix sum halves the ceiling. And it *closes off the obvious escape route*: the ceiling rises with window radius, but a window is vacuous once it spans the sequence, so $2w{+}1 \leq T$ must hold, and at speech-command temporal resolution no configuration, existing or hypothetical, exceeds 5.33%. We compare like with like throughout. Proposition 1 bounds the operation-count term, whereas end-to-end energy additionally reflects the firing-rate reduction inside the routed layers, which the bound does not model and which is second-order but not zero; forcing a trained gate from 0% to 100% skip moves the operation count by 7.1% and the measured energy by 10.9% (§2.5), and wherever we compare against a reported saving we use the measurement rather than the bound, since it is the larger of the two.

Where the reported savings actually come from. The reductions are real; the attribution is not. Measuring energy at every rung of an ablation ladder that adds one change at a time separates the contributions (§3.2). Narrowing the operator to half width accounts for -37.3 pp on SHD-1L and -21.1 pp on SSC-2L, while adding the learned gate on top accounts for $+2.8$ pp and -1.6 pp respectively: on SHD the gate costs marginally more than it saves. A further 8.0 to 17.5 pp of the published figure is not a property of the model at all but of the energy formula used to evaluate it. Recomputed correctly, the method's own reduction is 34.5% and 22.7% rather than 49.5%, and essentially all of it is the width reduction. There is an irony in the pairing. The width bottleneck is the change that delivers the saving, and it is simultaneously the change that halves ρ and caps what routing can add. The static intervention that pays is the one that caps the dynamic one.

Why a ceiling this simple went unnoticed. The explanation appears to be instrumental rather than conceptual. We document four defects in standard measurement practice, all found in our own implementation and all biasing efficiency claims in the same direction (§4): module-hook operation counts omit the tensor operations conditional computation targets, training-mode routing statistics differ from inference behaviour by a factor of seven here, an unweighted layer-mean firing rate overstates the routing effect by 4.3 to $4.8\times$, and reporting the best epoch without a validation split is model selection on test, worth 2.1 to 3.5 points in our data.

The constructive half. The ceiling is set not by how well a router learns but by where it is attached. The channel MLP holds 49 to 81% of the energy budget in these models, more than attention in every configuration, and a router that removes a frame from the whole block rather than only from the attention operator faces a ceiling 4.6 to $28.8\times$ higher (§2.4). We then reach that ceiling. Holding the routing rule, rate and training protocol fixed and varying only where the mask is applied, one random rule removes 2.7% of energy at the attention operator, 11.0% at the channel MLP and 25.9% at the whole block, against ceilings of 2.83, 12.2 and 29.8%, with accuracy flat to within a point (§3.4). The bound predicts all three across an order of magnitude, and what differs between them is not the routing rule, the recipe, or the quality of a learned decision, since none of those varies. It is which operator the mask is applied to.

Contributions.

1. **A ceiling on query-side conditional attention**, $\Delta C/C \leq u_{\mathrm{id}} s_{\mathrm{attn}} \rho$ on routed operations, with a decomposition of ρ into receptive-field locking and shared projections, the second dominating the first two-to-one here (§2.1).

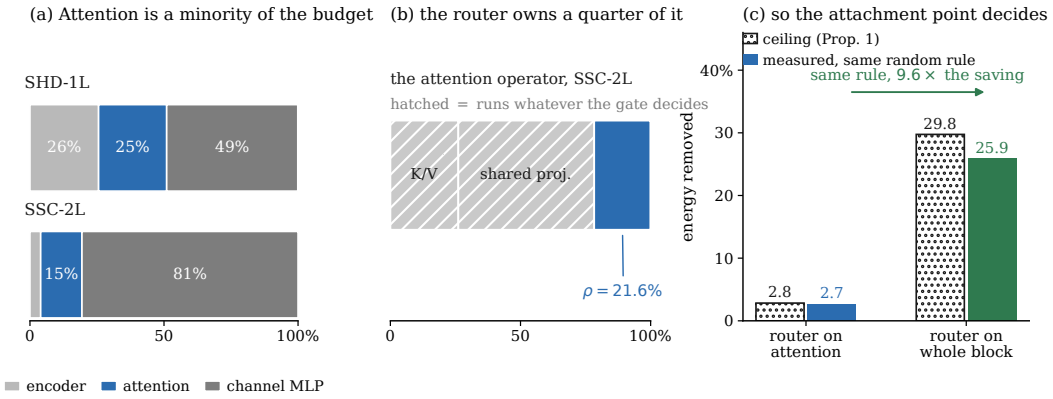

Figure 1: The whole argument. **(a)** Attention is a minority of the energy budget in every speech configuration we measured, and the channel MLP is the plurality. **(b)** Inside the attention operator, most of the cost is unavailable to a router: keys and values are consumed by neighbouring queries, and the bracketing projections run per frame regardless of the routing decision, leaving $\rho = 21.6\%$ on SSC-2L. The ceiling is essentially the product of (a) and (b). **(c)** Consequently the attachment point, not the router, decides what is achievable: the *same* i.i.d. random routing rule at the same rate removes 2.7% of energy at the attention operator and 25.9% at the block, each within a few points of the ceiling we derive for it, and the block-level arm scores above the un-routed baseline.

2. **An evaluation over seventeen configurations from eight published architectures** under one formula and adversarial assumptions, with an infeasibility result: at speech-command resolution the profitable region does not exist (§2.2).
3. **An attribution of the reported savings** by measuring energy at every rung of a six-rung ablation ladder under both accountings, locating the saving in a static width reduction and a measurement artefact rather than in the gate (§3.2).
4. **A matched-rate comparison** over three seeds in which a learned gate trails a coin flip at the same rate by 0.08 points ($p = 0.40$) (§3.1, §3.3).
5. **A reachability result**: with the routing rule fixed and only the attachment point varied, block-level routing recovers 25.9% of energy at accuracy above the un-routed baseline (§3.4).
6. **Four measurement pitfalls**, each quantified on our own runs, each with a remedy and a cheap diagnostic, released as instrumented tooling (§4).

2 A Ceiling on Query-Side Conditional Attention

This section is self-contained arithmetic: no experiment in it required training a model.

2.1 Setting and the Bound

A spiking transformer unrolls over T timesteps and maps N frames through an input encoder, L blocks each pairing an attention operator $\mathcal{A}$ with a channel MLP $\mathcal{M}$, and a linear head. Cost is measured in synaptic operations: accumulates for spike-driven layers, multiply–accumulates for the single real-valued input layer (Horowitz, 2014; Lemaire et al., 2023). Write $C = C_{\mathcal{E}} + L(C_{\mathcal{A}} + C_{\mathcal{M}}) + C_{\text{head}}$, with $s_{\text{attn}} := LC_{\mathcal{A}}/C$ and $s_{\text{mlp}} := LC_{\mathcal{M}}/C$; the repeat factor T multiplies encoder and blocks alike and cancels from every ratio below.

What we count as attention. $C_{\mathcal{A}}$ includes everything between the residual-stream taps: the $Q/K/V$ projections, the similarity and aggregation, any sliding-window accumulation, the spiking gate and the projections bracketing the operator. This is deliberately broad, because it is the boundary a router attaches to; it is wider than counting $QK^{\top}$ alone, and Appendix J reconciles the two conventions against a measured per-layer count.

Definition 1 (Query-side routing)**.** *A router emits $m_t \in \{0, 1\}$ per frame. Under* query-side *routing, $m_t = 0$ suppresses the computation producing frame t's own attention output, while frame t's key and value are still materialised because neighbouring queries consume them. Mixture-of-experts blocks with an identity or skip expert are of this form, as are attention-level early-exit and skip gates.*

Table 1: The eight audio configurations: attention share, query-side fraction, and the ceiling of Proposition 1 at a 40% routing rate and in the degenerate limit where attention never executes. No windowed architecture exceeds 7% at a realistic routing rate, and the two rows that do use global attention. The nine vision configurations retained as a control span 5.61 to 6.92% and are listed in Appendix I.

Model	D	L	N	s_{attn}	s_{mlp}	ρ	$\bar{\Delta}_{40}$	$\bar{\Delta}_{100}$
SpikCommander SHD-1L	128	1	100	25.4%	48.9%	27.8%	2.83%	7.07%
SpikCommander SSC-1L	256	1	100	14.8%	77.4%	21.6%	1.28%	3.21%
SpikCommander SSC-2L	256	2	100	15.4%	80.6%	21.6%	1.33%	3.34%
SpikCommander GSC-1L	256	1	101	14.8%	77.5%	21.6%	1.28%	3.21%
SpikCommander GSC-2L	256	2	101	15.4%	80.6%	21.6%	1.33%	3.34%
SpikCommander SHD-1L (alpha=4)	128	1	100	14.7%	70.6%	27.8%	1.63%	4.08%
Spikformer-audio (SSC/GSC 2L)	256	2	100	37.2%	62.2%	58.2%	8.65%	21.63%
IML-Spikeformer (ASR enc.)	256	12	250	42.6%	57.3%	66.4%	11.32%	28.31%

Proposition 1 (Ceiling on routed operations)**.** *Let $\rho \coloneqq C_{\mathcal{A}}^{\mathrm{query}}/C_{\mathcal{A}}$ be the query-side fraction of the attention operator and u_{id} the expected fraction of frames routed away. Then*

$$\frac{\Delta C}{C} \;\leq\; u_{\mathrm{id}} \cdot s_{\mathrm{attn}} \cdot \rho. \tag{1}$$

Proof. Routing removes no operations outside $\mathcal{A}$, so $\Delta C \leq u_{\mathrm{id}} L C_{\mathcal{A}}^{\mathrm{query}}$. Dividing by C and substituting $LC_{\mathcal{A}}^{\mathrm{query}} = \rho\, s_{\mathrm{attn}} C$ gives (1). The bound is loose in the model's favour: it charges nothing for the gate and assumes the routed computation is perfectly elided. □

From operations to energy, and what we do not bound. The proposition is deliberately about *operations*. Energy here is $\sum_l r_l u_l F_l$, so routing moves it through two channels: it removes operations inside the gated operator, which (1) bounds, and it changes the firing rates of the layers downstream, which (1) does not model. We do not bound the second, since that would require fixing an activation statistic, and it is not small: skipping everything removes 7.1% of operations but 10.9% of energy on SHD-1L. Two consequences hold throughout: every energy claim here is a *measurement* rather than a corollary of the proposition, and wherever we compare against a recorded saving we use the measured figure, which is the larger.

The literature has examined the first factor by designing better routers and has implicitly assumed the second is large. The third has gone unexamined, and it is the one that binds.

Why $\rho < 1$, and what it decomposes into. Two mechanisms keep parts of the operator alive regardless of the router's decision, both shown in Figure 1(b). *Receptive-field locking*: frame t's key and value are read by every query whose window contains t, which in a fused QKV projection pins two thirds of it at full rate. *Shared projections*: any projection bracketing the operator, in particular a width bottleneck $D \to D_e \to D$, runs per frame whatever the router decides, because the identity path's output must be projected back too. For sliding-window speech models the split is exactly 1:2, and it is parameter-free, which is why it repeats to the digit across configurations: with $D_e = D/2$ the per-frame costs are D_e^2 for Q, $2D_e^2$ for K and V jointly, and $4D_e^2$ for the two bracketing projections. Naming either mechanism alone would misdescribe the bound, and for global-attention models the second term vanishes entirely.

2.2 Seventeen configurations, one formula

We evaluate (1) on eight audio configurations, our primary evidence base, and on nine vision configurations retained as a control (Appendix I, which also lists every value, its source and every assumption). Three rules govern the table. **R1**: one formula for every row, ours included. **R2**: where a paper omits a parameter we adopt the value *favourable to attention*, that is, adversarial to our conclusion, and flag it. **R3**: windowed attention is reported under two implementations (§2.3).

Attention is neither dominant nor negligible. s_{attn} spans 14.7 to 42.6% while the channel MLP spans 38.3 to 80.6% and is the plurality of the budget in every audio configuration. This contradicts the premise stated in this line of work that attention accounts for the majority of per-sample energy, and equally contradicts dismissing it as negligible; the intermediate statement is the accurate one,

and it is what makes ρ decisive rather than s_{attn}. Meanwhile $\bar{\Delta}_{40}$ lies in $[1.28\%, 11.32\%]$ over all seventeen configurations and in $[1.28\%, 2.83\%]$ for the five streaming-capable speech ones; the two audio rows above 8% use global rather than windowed attention and neither can stream. Locality is not incidental to speech deployment but required by it, and it is one of the two mechanisms that suppress ρ in these models.

2.3 Three attempts to escape the bound

Three responses are available to a designer who finds the ceiling too low, and none of them works; Appendix C gives each in full. *Remove the bottleneck*: two thirds of the irreducible cost comes from the $D \to D/2 \to D$ projections, which exist only because the method we examine narrows its experts, so a reader may object that the ceiling follows from a design choice inside the method being criticised. Recomputing with experts at full width and no bracketing projections raises ρ to 39.9 to 45.3% and the ceiling to 3.71 to 6.80%: it nearly triples and stays an order of magnitude below the reported savings. This exposes a real trade-off, since the width bottleneck is the change that delivers most of the reported saving and is simultaneously the change that halves ρ. *Improve the implementation*: replacing the materialised window with a running prefix sum, which is what a deployment would use, is entirely query-side and therefore *lowers* ρ, taking the ceiling from 2.83% to 1.39% on SHD-1L. *Widen the window*: the ceiling grows with w, but w is bounded by the sequence, since once $2w{+}1 > T$ the window spans everything, so $w \leq T/2$. Under that constraint, at the $T = 100$ used by every benchmark here, the most favourable feasible point is $D = 64$, $w = 50$ at $\bar{\Delta}_{40} = 5.33\%$, and at the widths actually used the feasible maximum is 3.21% and 2.05%. The claim is therefore not that no published model occupies the profitable region but that within this family it does not exist at this temporal resolution. Two qualifications keep the scope honest. The sweep varies (D, w) with the rest of the geometry fixed at the windowed, bottlenecked form, so it bounds that family rather than conditional attention in general: the no-bottleneck variant above reaches 6.80%, outside the 5.33% maximum precisely because it is outside the swept family. And the constraint is resolution-specific, since at $T = 1000$ the feasible maximum rises to 23.2%.

2.4 Where the bound does not apply: block-level dropping

Proposition 1 constrains query-side routing only. Mixture-of-Depths (Raposo et al., 2024) and token-dropping remove a frame from the block entirely: its keys and values are never formed and its MLP never runs, so neither mechanism behind ρ applies. Recosting each block at $N' = N(1 - u)$ gives ceilings of 26.7 to 43.3% at the same routing rate, a factor of 3.8 to 28.8 above query-side routing; Appendix I gives the figure for every configuration.

The ordering inside that comparison is the paper's constructive finding. Our SSC-2L configuration has the worst query-side ceiling, 1.33%, and the best block-level ceiling, 38.4%, for the same reason in both cases: its channel MLP holds 80.6% of the budget, which query-side attention routing cannot reach and block-level routing can. We therefore read the analysis as a design criterion. Conditional computation in spiking speech models is bounded by its attachment point, and the field has converged on the attachment point with the lowest ceiling. Whether the higher ceiling is *reachable* in a sliding-window model, where dropping a frame also removes it from its neighbours' context, is an empirical question we answer in §3.4. Finally, a backstop: even the block-level ceiling of 38.4% remains below the 49.5% reduction reported for query-side routing on SHD-1L, so the central claim of this paper does not depend on the streaming argument surviving.

2.5 Measuring the ceiling on a trained model

The bound so far is arithmetic. We confirm it directly, forcing a trained checkpoint's gate to skip a prescribed fraction of frames at inference and measuring operations and energy at each rate, with no training involved and one evaluation pass per point. The operation count falls by 2.9% at a 40% skip rate and by 7.1% when attention never runs, matching Proposition 1 to within the slack noted in its proof, and measured energy falls further, by 5.1% and 10.9%.

The gap between 7.1% and 10.9% is the unmodelled firing-rate channel of §2.1, measured rather than assumed: not a violation of a bound on operations, but not negligible.

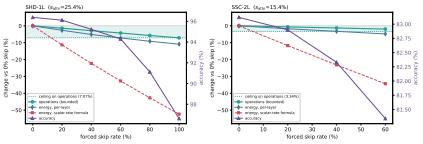

Figure 2: Forcing a trained model's gate across the whole skip axis at inference. Proposition 1 bounds the *operation count*, and the measured operation curve tracks that bound (shaded). Energy falls somewhat further, because skipping the expert also lowers the firing rate of the layers downstream of it, a second-order term the bound does not model. Both are an order of magnitude below the dashed curve, which is the same measurement under the scalar-rate formula of §4. Accuracy, on the right axis, falls by nearly ten points across the same sweep.

Repeating the sweep on SSC-2L, whose attention share is 15.4% against SHD-1L's 25.4%, checks that the bound scales as it should: the measured operation count falls by 1.29% against a predicted 1.33%, nearly tight, where on SHD-1L the same quantities are 2.9% and 2.83%. On both the bound is respected and close, and on both the energy curve falls further than it. That is the whole result in one picture (Figure 2): *the energy curve is flat to within a few percent while the accuracy curve falls by ten points.* The same sweep under the scalar-rate formula in common use reports -52.3% instead of -10.9%, a discrepancy we take up in §4.

2.6 Where the budget actually is

The channel MLP holds more of the budget than attention does in every speech configuration we measured, 48.9% on SHD-1L and up to 80.6% on SSC and GSC, and it is tempting to conclude it is pointwise in time and so free to gate with $\rho \approx 1$. It is not: a depthwise convolution with kernel 31 along the time axis runs over half its expanded channels, so a routed-away frame must still supply its neighbours' windows. Working the cost through (Appendix D) gives $\rho_{\mathrm{mlp}} = 62.5$ to 70.0% and ceilings of 12.22% on SHD-1L and 22.57% on SSC-2L, against 2.83% and 1.33%: where attention-side routing is most hopeless, MLP-side routing is most promising, because an MLP absorbing four fifths of the budget causes both. A *learned* gate cannot test this, and the reason is itself a result: moved to the channel MLP, the same gate routes 100% of frames away within a dozen epochs and the network still reaches 92.0% validation accuracy on SHD with no channel MLP at all. A collapsed gate measures the value of *deleting* an operator rather than of *routing* it, so the attachment-point comparison holds the routing rule fixed instead, as §3.4 does.

3 Attributing the Reported Savings

Section 2 bounds what routing can remove; this section asks whether the recorded savings separate into a routing term and static terms, and whether a learned gate buys anything over a random one at a matched rate. Protocol is in Appendix K.

3.1 An ablation ladder that adds one change at a time

The block we examine bundles three changes relative to its baseline attention operator: removal of the long-range branch, a halving of the operator width, and the routing gate itself. A six-rung ladder separates them. **A** is the baseline operator, sliding-window and long-range branches at width D; **B** keeps the sliding-window branch only at width D; **C** narrows it to $D/2$; **D** adds i.i.d. random routing at the rate the learned gate converges to, isolating *doing less work* from *learning what to skip*; **E** substitutes a non-learned threshold on per-frame spike count; and **F** is the full method. Rungs D and E are new, neither exists in the original method, and both are required before anything can be attributed to learning; both are calibrated to the rate F converges to *at inference* rather than to the rate reported in its training log (§4).

Accuracy, and where rung B sits. Differences on this ladder are frequently below one point, so accuracy is reported over three seeds on SSC-2L with true test accuracy: A at 83.34 ± 0.22, C at 82.86 ± 0.35, D at 82.90 ± 0.16 and F at 82.82 ± 0.23. Paired across seeds, the learned gate trails random routing at the same rate by 0.08 points ($p = 0.40$), the un-routed rung by 0.04 points ($p = 0.90$), and the baseline operator by 0.52 points ($p = 0.15$). None is detectable at this power, and that null is the premise the energy attribution rests on: no change on this ladder buys accuracy,

Table 2: Energy per rung under both accountings, relative to the baseline attention operator (rung A). The static changes deliver the reduction; routing delivers approximately nothing, and on SHD it costs slightly more energy than it saves. The gap between the corrected and legacy columns is the measurement artefact of §4. Accuracy is reported separately in §3.1 and §3.3 under the leak-free estimator, so that a single estimator governs every accuracy number in this paper.

	SHD-1L			SSC-2L		
Rung	E (mJ)	correct	legacy	E (mJ)	correct	legacy
A baseline attention	0.002342	—	—	0.019101	—	—
B SWA only, D	0.002399	$+2.4\%$	-10.0%	0.019802	$+3.7\%$	-13.5%
C SWA only, $D/2$	0.001469	-37.3%	-32.0%	0.015078	-21.1%	-22.4%
D C + random	0.001429	-39.0%	-46.5%	0.014429	-24.5%	-41.3%
E C + spike-count	0.001480	-36.8%	-51.4%	0.016473	-13.8%	-39.9%
F full method	0.001534	$\mathbf{-34.5}\%$	$\mathbf{-42.5}\%$	0.014770	$\mathbf{-22.7}\%$	$\mathbf{-40.2}\%$

so the energy differences below are not traded against it. Rung B is reported at a single seed, which we flag rather than drop, because B separates the two static changes from each other and because operation counting is near-deterministic, so the energy table carries it at full weight.

3.2 Energy at every rung, under both accountings

The accuracy ladder tells us what each change costs. The question this paper opened with is what each change *saves*, answered by measuring energy at every rung under both the corrected per-layer accounting of §4 and the scalar-rate formula in common use. This separates two candidate explanations for the reported 49.5% that we have so far kept apart: that it is dominated by the static changes (A→C), or by a measurement artefact (the gap between the corrected and legacy columns at rung F).

Table 2 answers it, and the answer has two parts rather than one.

The static changes deliver the reduction. Narrowing the operator to half width (A→C) accounts for -37.3 pp on SHD-1L and -21.1 pp on SSC-2L. Removing the long-range branch (A→B) costs slightly *more* energy under correct accounting, because the surviving branch is widened to compensate. The saving comes from the narrowing, not from the branch removal, which is why rung B has to be in the table at all.

Routing delivers approximately nothing, exactly as the ceiling predicts. Adding the learned gate to the narrowed operator (C→F) changes energy by $+2.8$ pp on SHD-1L, where the gate costs marginally more than it saves, and by -1.6 pp on SSC-2L. Proposition 1 predicts at most 0.9% for SSC-2L at this gate's measured inference-mode routing rate, and the measured 1.6% sits just above it for the second-order reason set out in §2.5. The bound is confirmed at the level of a single percentage point on a trained model, not merely on paper.

The measurement artefact is the remainder. Under the scalar-rate formula the same method reports -42.5% and -40.2% instead of -34.5% and -22.7%, an inflation of 8.0 and 17.5 pp respectively. The distortion is not a constant factor and does not cancel between architectures. Most starkly, the A→B comparison changes *sign*, from $+2.4\%$ under correct accounting to -10.0% under the legacy formula. A reader using that formula would conclude that removing the long-range branch saves energy, when measurement shows it costs energy.

The method's own number, recomputed. Putting the two together: recomputed with per-layer accounting, the full method's energy reduction relative to the baseline attention operator is 34.5% on SHD-1L and 22.7% on SSC-2L, against the 49.5% reported. Of the 34.5%, essentially all is attributable to the width reduction and none to conditional computation.

3.3 Matched routing rates

A comparison between a learned gate and a random one is only informative if both route at the same rate, and the published configuration compares them at different rates, which favours whichever arm

Table 3: One routing rule, three attachment points, on SHD-1L with the leak-free estimator. The ceiling predicts each arm to within a few points and the measured ordering is the predicted one, while accuracy is flat. Varying the block-level rate instead gives -2.1% at 20% and -46.2% at 60%.

Router attached to	Rate	Acc. (%)	E (mJ)	ΔE vs C	ceiling
— (rung C, no routing)	—	91.79	0.001469	—	—
attention operator only	41%	93.36	0.001429	−2.7%	2.83%
channel MLP only	40%	93.01	0.001307	−11.0%	12.2%
whole block	40%	93.77	0.001089	**−25.9%**	29.8%

does more work. On SSC-2L, where the training-mode and inference-mode averages agree at 27%, random routing at that rate scores 82.74% against the learned gate's 82.88% on the same seed, a gap of 0.14 points well inside the seed-to-seed spread of §3.1. We put this no more strongly than the data allows: this is one paired seed, and the arms match on the network-average routing rate but not on its distribution over depth, since the learned gate routes about 5% of frames away in the first block and 49% in the second where the random arm routes 27% in each (Appendix E). No advantage for the learned gate was detected, which is weaker than equivalence and is what we claim. Our first attempt at the SHD arm was not matched, because we calibrated against the 27% rate in the training log while the same checkpoint routes 3.6% at inference, a factor of seven and an instance of §4; those arms nonetheless bound the answer in the same direction, since each skips seven to eleven times more frames than the learned gate does at inference and none loses more than 1.35 accuracy points against it (Appendix K). The learned gate is not buying its accuracy by routing well. It is very nearly not routing at all.

3.4 Reaching the higher ceiling: varying the attachment point

Section 2.4 shows that removing a frame from the entire block raises the ceiling by 4.6 to $28.8\times$. Whether that ceiling is *reachable* in a sliding-window model is a separate question, since a dropped frame is also removed from its neighbours' receptive fields, which is exactly the context a streaming speech model depends on. We answer it empirically rather than by assertion, implementing full-drop routing at three rates, under which a dropped frame's key and value are never formed and its channel MLP never runs at all.

The higher ceiling is real and reachable. At a 40% drop rate, block-level routing removes 25.9% of the energy against the 29.8% that §2.4 predicts, while scoring 93.77%, *above* the no-skip rung C and within 0.2 points of the learned query-side gate; at 60% it removes 46.2% and still scores above C, whereas query-side routing at a comparable rate removes 2.7%. This is the paper's constructive result: the difference between a 2.7% saving and a 25.9% saving is not the routing rule, the training recipe, or the quality of the learned decision, since both arms use the same random rule at similar rates. It is which operator the mask is applied to. We had expected the opposite, because removing a frame from a sliding-window model also removes it from its neighbours' receptive fields, and §2.4 anticipated that this would cost enough accuracy to make the higher ceiling unreachable. It does not, and we report this because it contradicts what we wrote before running it.

What this arm does and does not isolate. Accuracy rising above the un-routed rung is the part to be careful with. Stochastic frame dropping is also a regulariser, so the accuracy column cannot separate a benefit of the attachment point from a benefit of the training noise, and doing so needs a control we have not run. What the arm isolates is the *energy*, a property of the computation rather than of how it was trained, measured with the routing rule and rate held fixed and only the attachment point varied. We read Table 3 as evidence about where a router can reach, not yet that block-level routing improves accuracy; the arm is single-dataset and single-seed, and its run-to-run variability is recorded in Appendix F.

4 Four Measurement Pitfalls

The ceiling of §2 is elementary arithmetic; that it went unnoticed is not, and the explanation appears instrumental. We found four defects in our own implementation, all biasing efficiency optimistically and all, we believe, common. **P1**: module hooks miss the window accumulation, spiking gate and modulation, which live in no `nn.Module` and are exactly what a query-side router skips, amounting

here to 3.8% of network operations but 57% of everything routing can touch. **P2**: the gate's batch normalisation makes its pre-activation law mode-dependent, and we log 27% identity usage where inference gives 3.6%. **P3**: an unweighted layer-mean firing rate applied to a network total lets a change confined to small layers rescale the whole model, reporting -52.3% where per-layer accounting reports -10.9%. **P4**: taking the maximum over epochs without a validation split is model selection on test, worth 2.1 to 3.5 points here. The four compound, since each independently favours the conclusion that the intervention works, so a method evaluated with all four reports a favourable energy–accuracy trade-off whether or not it has one. These are defaults rather than lapses, and the strongest evidence is our own: while writing this paper we recorded a 25.2% saving that is 5.5% under correct accounting. Two of the four are also *configuration-dependent*, which bounds what can be concluded from them: on GSC-2L the two routing statistics agree closely and the scalar-rate formula inflates the routing effect by $1.00\times$ rather than 4.3 to $4.8\times$. A practice that distorts one configuration can leave another untouched, so a defect in a script establishes that a number *may* be wrong, never that it is. Appendix H gives each with a remedy.

5 Related Work

Spiking transformers, and conditional computation. Spikformer (Zhou et al., 2023) introduced spiking self-attention by removing the floating-point softmax, and successors have refined the operator: linear-complexity spike-driven attention (Yao et al., 2024), spatio-temporal attention (Lee et al., 2025), addition-only similarity (Guo et al., 2025), α-XNOR similarity (Xiao et al., 2025) and spike-timing-based token prioritisation (Jang, 2025). For speech, SpikCommander (Wang et al., 2026) combines sliding-window and long-range spike-temporal attention with a contextual-refinement channel MLP, and IML-Spikeformer (Song et al., 2026) scales spiking transformers to full ASR. Our analysis takes these architectures as given and asks what fraction of their energy budget is reachable by a router, an answer that does not depend on which similarity operator they use. Sparsely-gated mixture-of-experts (Shazeer et al., 2017) and its successors (Fedus et al., 2022; Lepikhin et al., 2021; Zhou et al., 2022) decouple capacity from per-input cost, conditional computation has a long history in efficient inference (Han et al., 2022; Scardapane et al., 2024), and in the spiking domain routing has been applied along spatial axes for vision (Zhou et al., 2024; Xu et al., 2024) and to individual neurons (Cheng et al., 2025). Mixture-of-Depths (Raposo et al., 2024) is the closest design to the alternative we evaluate, letting tokens skip entire layers rather than a single operator, which, as §2.4 makes precise, is the difference between a ceiling of 1.3% and one of 38.4% in the same architecture. Our contribution is not a new routing mechanism but a bound that applies to any of them once the attachment point has been fixed.

Efficiency, energy accounting, and re-examination results. Complementary work reduces spiking inference cost by regularising the firing rate (Yan et al., 2022), by pruning or quantisation (Liu et al., 2024; Wei et al., 2025; Castagnetti et al., 2025), and by early exit (Patne et al., 2026; Song et al., 2024). These act on different terms than routing, but under corrected accounting firing-rate regularisation reduces energy by 5.5% and routing by approximately nothing (Appendix G), both far smaller than the figures usually attached to them. Estimating energy from operation counts weighted by CMOS constants (Horowitz, 2014; Lemaire et al., 2023) is near-universal here, and recent work questions how well such estimates track real hardware (Yan et al., 2026); our concern is narrower, namely that the estimates are frequently not computed correctly even on their own terms. This paper belongs to the genre of re-examinations checking whether a reported improvement comes from the mechanism it is credited to, and we have tried to be at least as sceptical of our own analysis as of the work examined: every ambiguous parameter in Table 1 is resolved against our conclusion, §2.3 tests whether the bound is an artefact of a design choice inside the method we examine, and §3.4 reports a result contradicting the prediction we recorded beforehand.

6 Discussion and Conclusion

Proposition 1 bounds operations rather than energy and constrains query-side routing only, and a half-width single-branch attention operator remains the part of these methods that survives intact. Four scope conditions qualify the rest (Appendix B), and none touches the conclusion: the binding constraint is the attachment point.

Reproducibility statement

Every analytic table and figure is emitted by a single script from the parameters in Appendix, so no number here can drift from the code that produced it. The supplementary material contains that script, the instrumented counter, the routing probe, the leak-free evaluation script, and every routing mode.

AI use statement

We used a large language model to check and polish the writing of this paper, and to assist with code, specifically the plotting and figure-generation scripts. All experiments were run on our own hardware, and every number reported here is computed from stored test-set logits of models we trained and was verified by the authors against the underlying data files.

References

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

Andrea Castagnetti, Alain Pegatoquet, and Benoît Miramond. All in one timestep: Enhancing sparsity and energy efficiency in multi-level spiking neural networks, 2025. URL https://arxiv.org/abs/2510.24637.

Runxi Cheng, Yuchen Guan, Yucheng Ding, Qingguo Hu, Yongxian Wei, Chun Yuan, Yelong Shen, Weizhu Chen, and Yeyun Gong. Mixture of neuron experts, 2025. URL https://arxiv.org/abs/2510.05781.

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. 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, 2022. ISSN 2162-2388. doi: 10.1109/tnnls.2020.3044364. URL http://dx.doi.org/10.1109/TNNLS.2020.3044364.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Yufei Guo, Xiaode Liu, Yuanpei Chen, Weihang Peng, Yuhan Zhang, and Zhe Ma. Spiking transformer:introducing accurate addition-only spiking self-attention for transformer, 2025. URL https://arxiv.org/abs/2503.00226.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7436–7456, 2022. doi: 10.1109/TPAMI.2021.3117837.

Mark Horowitz. 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)*, pp. 10–14, 2014. doi: 10.1109/ISSCC.2014.6757323.

Minsuk Jang. Advancing sparse attention in spiking neural networks via spike-timing-based prioritization, 2025. URL https://openreview.net/forum?id=sL0NpgJRMs.

Donghyun Lee, Yuhang Li, Youngeun Kim, Shiting Xiao, and Priyadarshini Panda. Spiking transformer with spatial-temporal attention, 2025. URL https://arxiv.org/abs/2409.19764.

Edgar Lemaire, Loïc Cordone, Andrea Castagnetti, Pierre-Emmanuel Novac, Jonathan Courtois, and Benoît Miramond. An analytical estimation of spiking neural networks energy efficiency. In Mohammad Tanveer, Sonali Agarwal, Seiichi Ozawa, Asif Ekbal, and Adam Jatowt (eds.), *Neural Information Processing*, pp. 574–587, Cham, 2023. Springer International Publishing. ISBN 978-3-031-30105-6.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Yue Liu, Shanlin Xiao, Bo Li, and Zhiyi Yu. Sparsespikformer: A co-design framework for token and weight pruning in spiking transformer. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6410–6414, 2024. doi: 10.1109/ICASSP48485.2024.10446631.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks, 2019. URL https://arxiv.org/abs/1901.09948.

Parth Patne, Mahdi Taheri, Ali Mahani, Maksim Jenihhin, Reza Mahani, and Christian Herglotz. Sparq: Spiking early-exit neural networks for energy-efficient edge ai, 2026. URL https://arxiv.org/abs/2603.14380.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models, 2024. URL https://arxiv.org/abs/2404.02258.

Simone Scardapane, Alessandro Baiocchi, Alessio Devoto, Valerio Marsocci, Pasquale Minervini, and Jary Pomponi. Conditional computation in neural networks: Principles and research trends. *Intelligenza Artificiale*, 18(1):175–190, 2024. ISSN 2211-0097. doi: 10.3233/ia-240035. URL http://dx.doi.org/10.3233/IA-240035.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL https://arxiv.org/abs/1701.06538.

Zeyang Song, Qianhui Liu, Qu Yang, Yizhou Peng, and Haizhou Li. Ed-skws: Early-decision spiking neural networks for rapid,and energy-efficient keyword spotting, 2024. URL https://arxiv.org/abs/2406.12726.

Zeyang Song, Shimin Zhang, Yuhong Chou, Jibin Wu, and Haizhou Li. Iml-spikeformer: Input-aware multilevel spiking transformer for speech processing. *IEEE Transactions on Neural Networks and Learning Systems*, 37(3):1377–1389, 2026. doi: 10.1109/TNNLS.2025.3615971.

Jiaqi Wang, Liutao Yu, Xiongri Shen, Sihang Guo, Chenlin Zhou, Leilei Zhao, Yi Zhong, Zhiguo Zhang, and Zhengyu Ma. Spikcommander: a high-performance spiking transformer with multi-view learning for efficient speech command recognition. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i3.37194. URL https://doi.org/10.1609/aaai.v40i3.37194.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition, 2018. URL https://arxiv.org/abs/1804.03209.

Wenjie Wei, Malu Zhang, Zijian Zhou, Ammar Belatreche, Yimeng Shan, Yu Liang, Honglin Cao, Jieyuan Zhang, and Yang Yang. Qp-snn: Quantized and pruned spiking neural networks, 2025. URL https://arxiv.org/abs/2502.05905.

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

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

Yulong Yan, Haoming Chu, Yi Jin, Yuxiang Huan, Zhuo Zou, and Lirong Zheng. Backpropagation with sparsity regularization for spiking neural network learning. *Frontiers in Neuroscience*, Volume 16 - 2022, 2022. ISSN 1662-453X. doi: 10.3389/fnins.2022.760298. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2022.760298.

Zhanglu Yan, Zhenyu Bai, Kaiwen Tang, and Weng-Fai Wong. Reconsidering the energy efficiency of spiking neural networks inference from analytical perspectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2026. doi: 10.1109/TCAD.2026.3718799.

Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo XU, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1SIBN5Xyw7.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V Le, and James Laudon. Mixture-of-experts with expert choice routing. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=jdJo1HIVinI.

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

Zhaokun Zhou, Yijie Lu, Yanhao Jia, Kaiwei Che, Jun Niu, Liwei Huang, Xinyu Shi, Yuesheng Zhu, Guoqi Li, Zhaofei Yu, and Li Yuan. Spiking transformer with experts mixture. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 10036–10059. Curran Associates, Inc., 2024. doi: 10.52202/079017-0322. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/137101016144540ed3191dc2b02f09a5-Paper-Conference.pdf.

A Provenance of the Code and Measurements

This appendix records which parts of the software behind this paper we wrote, which parts we started from, and what we changed, so that a reader can separate our contribution from the starting point and can check the magnitudes quoted in §4 against specific changes rather than taking them on trust.

A.1 What is ours and what is not

The backbone is a published spiking speech transformer. We use its architecture, its released configurations and its public implementation as the starting point for every trained model in this paper. **We do not dispute, re-measure or re-report its accuracy results**, and nothing in this paper should be read as a claim about them. Table 1 treats that architecture exactly as it treats the sixteen others, as a set of parameters entering one formula.

Everything the paper actually examines is ours. The conditional-computation block, the six-rung ablation ladder, the four routing modes, the 49.5% figure, the recomputed 34.5% and 22.7%, the forced-skip sweep, and every accuracy and energy number in §3 come from our own implementation and our own runs. When §3.2 says that a reduction was misattributed, the reduction being reattributed is one we ourselves recorded, and the misattribution is ours.

A.2 The scope of the measurement claims

We state each of the four defects in §4 as a defect in *our* implementation, because that is where we found it and because every magnitude we quote for it was measured on our own runs. We draw

no conclusion from them about any number published elsewhere, and we want to be explicit about why the inference would not be valid even where a practice is widespread: a defect present in a measurement script does not by itself establish that a particular reported figure was produced with it, and we do not have the pipeline that produced any external figure. What we claim is narrower and sufficient for our argument. The practices we describe are in common use, they inflate this intervention by the margins we report, and under correct accounting the reduction we ourselves recorded falls from 49.5% to 34.5%, of which the routing mechanism contributes none.

A.3 Change log

The supplementary material contains a complete change log grouped into three categories. **Measurement fixes**: tensor-operation accounting alongside module hooks, the query-side and key/value split of the fused QKV projection, per-layer rather than layer-averaged SOP accumulation, block-level routing maps for the drop arm, inference-mode routing probes, a validation split for the benchmark that ships none, and a leak-free accuracy estimator. **New analysis tools**: the ceiling calculator that emits every analytic table in this paper, and the forced-skip sweep. **New routing modes**: random, spike-count, block-level drop, and the identity-bias control. Each entry names the file, states whether the code was inherited or written by us, and gives the effect of the change on the numbers reported here.

Two entries are worth flagging because they changed our own conclusions rather than merely our numbers. The per-layer SOP accumulation moved our recorded saving for firing-rate regularisation from 25.2% to 5.5%, which we discovered while already drafting §4 about that very defect. And the block-level routing map was added only after the drop arm's measured energy came out *above* the no-skip rung, which is impossible; without that check the constructive result of §3.4 would have been reported with the wrong sign.

B Limitations in Full

Proposition 1 constrains query-side routing at the attention operator and says nothing about routing attached elsewhere, nor about whether the accuracy of these models is good, which it is, nor whether their static architectural changes are worthwhile. Our reading is that they are: a half-width, single-branch spiking attention operator is a genuine contribution and is the part of these methods that survives our analysis intact. The infeasibility claim is tied to $T \approx 100$; at $T = 1000$ the feasible maximum rises to 23.2%, which is why our analysis points to long-sequence ASR rather than keyword spotting if conditional attention is worth pursuing in speech at all. Two rows of Table 1 rest on inferred parameters, and they are also the two least favourable to us, which is why we retain them; a sensitivity analysis over every plausible reconstruction (Appendix I) leaves ρ above 55.8% in all of them. Our energy figures are theoretical silicon estimates following the convention of this literature (Yan et al., 2026), a caveat that cuts both ways and does not affect our central comparison, which is between two accountings of the *same* model under the *same* convention. Accuracy differences here are frequently below one point, so we report effect sizes rather than significance claims; our trained experiments cover SHD and SSC, and the GSC rows are analytic only.

The attachment-point result does not separate placement from regularisation. The block-level arm of §3.4 scores above the un-routed rung. Stochastic frame dropping is also a regulariser, and a gain of that size is within what dropout-like noise can produce on this benchmark, so the accuracy column of Table 3 cannot attribute the gain to the attachment point. Two controls would separate them and we have not run either: the same dropping schedule applied without removing any computation, isolating the regularisation, and a static model matched to the drop arm's operation count, isolating the budget. The *energy* column is not subject to this confound, because it is a property of the computation performed rather than of how the model was trained, and it is measured with the routing rule and rate held fixed. The result we claim rests on that column.

The block-level arm is single-dataset and single-seed. It is run on SHD-1L only. Re-evaluating a saved checkpoint of a stochastic-routing arm draws a fresh Bernoulli mask and moves accuracy by up to 0.6 points, so all figures for those arms are means over the final ten epochs, and we attach no

ranking to differences at or below that scale. Replicating on SSC-2L with seeds is the first experiment we would add.

Matching an average rate is not matching a routing distribution. The one comparison in §3.3 that is matched on rate is matched on the network-average rate. The learned gate's inference-mode usage is not uniform across depth, routing about 5% of frames away in the first block and 49% in the second, where the random arm routes 27% in each. The arms therefore spend the same skipping budget differently over the network, and a 0.14-point difference between them does not distinguish *what* is routed from *where* the budget is spent. We accordingly claim only that no advantage for the learned gate was detected at this power and at this level of matching.

C THE THREE ESCAPE ROUTES IN FULL

Improve the implementation. The window accumulation may be materialised per position, at $\mathcal{O}(2w{+}1)$ each, as in the reference implementations we examined, or maintained as a running prefix sum at $\mathcal{O}(1)$ each, which is what a deployment would use. Because the accumulation is entirely query-side, improving it *lowers* ρ: the ceiling falls from 2.83% to 1.39% on SHD-1L and from 1.33% to 0.85% on SSC-2L. The better the attention implementation, the less there is for a router to save.

Widen the window. The ceiling grows with the window radius, since the accumulation is the only query-side term and scales with w. But w is bounded by the sequence itself: once $2w{+}1 > T$ the window spans everything and further growth is vacuous, so $w \leq T/2$. Reading the (D, w) sweep under that constraint closes the question. At $T = 100$, the resolution used by every benchmark here, the most favourable feasible point is $D = 64$, $w = 50$, giving $\bar{\Delta}_{40} = 5.33\%$; at the widths actually used the feasible maximum is 3.21% and 2.05%. The statement is therefore not that no published model occupies the profitable region, but that at this temporal resolution the region does not exist. The constraint is resolution-specific, and saying so keeps the argument honest: at $T = 1000$ the feasible maximum rises to 23.2%, which is why the two global-attention ASR encoders sit above the speech-command models. Appendix I plots the region.

D THE CHANNEL MLP AS AN ATTACHMENT POINT

In every speech configuration we measured, the channel MLP holds more of the energy budget than attention does: 48.9% on SHD-1L and 77.4 to 80.6% on SSC and GSC. It would be convenient to conclude that the MLP is pointwise in time and therefore free to gate, with $\rho \approx 1$, but that is wrong for the module used in these models, and getting it wrong would have inflated the prescription we make. The spiking contextual-refinement MLP applies a *depthwise convolution with kernel* 31 *along the time axis* to half of its expanded channels, so when frame t is routed away, the first pointwise convolution and the convolved half of the expanding map must still run to supply the depthwise windows of t's neighbours: the same receptive-field locking that suppresses ρ at the attention operator. Per frame the block costs $2D^2 + 2rD^2$, of which $D^2 + \frac{3}{2}rD^2$ is query-side, giving $\rho_{\mathrm{mlp}} = 62.5\%$ at $r = 1$ and 70.0% at $r = 4$: well below one, but two to three times the value at attention. The resulting ceilings are 12.22% on SHD-1L and 22.57% on SSC-2L, against 2.83% and 1.33% at the attention operator. The ordering is the constructive claim: the configuration where attention-side routing is most hopeless is where MLP-side routing is most promising, because an MLP absorbing four fifths of the budget causes both.

A learned gate cannot be used to test this, and the reason is itself a result. Moving the same learned gate to the channel MLP does not survive training: within a dozen epochs it routes 100% of frames away and stays there, and the load-balancing term at the weight used for the attention-side gate does not hold it back. The resulting network is a spiking transformer with attention, residual connections and no channel MLP at all, and it still reaches 92.0% validation accuracy on SHD after eighteen epochs, against roughly 93 to 96% for fully trained models with the MLP present. A collapsed gate measures the value of *deleting* an operator rather than of *routing* it, so the attachment-point comparison must hold the routing rule fixed, which is what §3.4 does.

Table 4: Matched-rate comparison on SHD-1L, leak-free estimator. The rate the non-learned arms were matched to is the learned gate's *training-mode* identity usage; its inference-mode usage on this dataset is 3.6%, so the comparison is not matched in the way we intended. We report it as such.

Arm	Routing rate	Acc. (%)	vs F
F learned gate	27% train / 3.6% eval	93.98	—
D random	41%	93.36	−0.62
E spike-count gate	41%	93.28	−0.70
D_{27} random	27%	93.48	−0.50
E_{27} spike-count gate	27%	92.63	−1.35

E Matched Routing Rates in Full

§3.3 reports the one comparison that is matched as designed. This appendix records the arms that are not, and why, because the failure is instructive and is an instance of §4 rather than a separate mistake.

The SHD arms are matched to the wrong rate. We built the 27% arms to match the rate at which the learned gate routes, read from the training log. Measured in inference mode on the test split, which is the number describing what a deployed model actually does, the same checkpoint routes 3.6% of frames to the identity expert, a factor of seven, and we did not notice until we measured the checkpoint rather than reading the log. Every non-learned arm in the table therefore skips seven to eleven times more frames than the learned gate does, which biases the comparison *towards* the learned gate. That none of them loses more than 1.35 accuracy points against it, while doing substantially less work, is therefore a bound in the direction that matters, but it is not the matched comparison we set out to run.

Matching an average rate is not matching a routing distribution. Even the SSC-2L comparison, where the training-mode and inference-mode averages agree at 27%, is matched only on that average. The learned gate's inference-mode usage is not uniform across depth: it routes about 5% of frames away in the first block and about 49% in the second, whereas the random arm routes 27% in each. The two arms therefore allocate the same total skipping differently over the network, and a difference of 0.14 accuracy points between them does not distinguish *what* is routed from *where* the routing budget is spent. A per-layer matched arm, in which the random rule reproduces the learned gate's per-block rates rather than its network average, is the experiment that would settle this and we have not run it. We state the conclusion accordingly in §3.3: no advantage for the learned gate was detected at this power and at this level of matching, which is weaker than a claim of equivalence and is all the data supports.

F Block-Level Dropping: Additional Measurements

Energy does not fall linearly with the drop rate. The 20% row is off the trend of the other two, and the reason is visible in the firing rates. Dropping 20% of frames removes 10.7% of operations, but the network's mean firing rate *rises* from 0.176 to 0.184, cancelling most of the saving; at 40% and 60% the rate falls instead, to 0.163 and 0.142, and the savings appear. A network trained under mild dropping appears to compensate by firing more, and under heavy dropping it cannot. Operation counts and energy do not move together in spiking models, and reporting one as a proxy for the other is unsafe in either direction.

Two measurement notes. The operation counter had to be extended for this arm, because under block-level dropping the sibling channel MLP is masked as well, which an expert-based routing map does not see; before the fix the drop arm's measured energy came out *above* the no-skip rung, which is impossible, and we flag it because it is the same class of error as the four in §4, found in our own tooling. Separately, arms with stochastic routing have stochastic inference, so re-evaluating a saved checkpoint draws a new mask and moves accuracy by up to 0.6 points; all comparisons here use the mean over the final ten epochs rather than a single evaluation.

Table 5: Four defects in standard measurement practice, the magnitude of each on our own runs, and a cheap diagnostic for it. Each error is one-sided in the optimistic direction, and each is invisible in a static comparison of two architectures while becoming first-order exactly when the intervention is confined to a single operator.

Defect	Why it is wrong	Magnitude here	Diagnostic
P1 Module-hook op counting	Window accumulation, spiking gate and modulation live in no `nn.Module`, so hooks miss them, and they are exactly what a query-side router skips	3.8% of network ops, but 57% of everything routing can touch	A counter whose total is invariant to the window radius is not counting the window
P2 Training-mode usage	The gate has batch norm behind dropout, so its pre-activation law differs by mode and a hard $\arg\max$ moves mass across the boundary	27% logged against 3.6% at inference; sign differs between blocks ($0.16/0.38$ vs $0.05/0.49$)	Recompute usage with `model.eval()` on held-out data before any energy claim
P3 Unweighted layer-mean firing rate	Rate is per-layer and must pair with that layer's op count; an unweighted mean lets a change in small layers rescale the whole model	-52.3% against -10.9% on the same sweep; $+8.0$ and $+17.5$ pp on the full method; reverses one sign	Force the rate to 100% and check the reduction does not exceed the routed component's share of ops
P4 Best-epoch accuracy without a validation split	Maximum over epochs of test accuracy is model selection on test, an order statistic growing with epochs and curve variance	2.1 to 3.5 points, not constant across configurations; inverts the ordering of two rungs	Report a fixed-epoch or last-k mean, which needs no retraining, or carve a validation split

G Routing against Firing-Rate Regularisation

Methods in this line describe routing and firing-rate regularisation as orthogonal axes that compose. Measured on the same model under corrected accounting the two do compose, with regularisation at $\lambda = 10^{-3}$ reducing energy by a further 5.5% at a cost of 0.75 accuracy points, but the magnitudes are not what the usual framing suggests: regularisation contributes 5.5% and routing approximately nothing, so presenting them as comparable axes overstates the second rather than the first. Under the scalar-rate formula this same comparison yields -25.2%, which is the number we initially recorded before correcting the accounting, while already writing a paper about that defect.

H The Four Pitfalls in Full

This appendix expands §4. Each pitfall is stated as: what the practice is, why it is wrong, how large the error is in our own measurements, and what to do instead. Every magnitude quoted here is measured on our runs, and every remedy is implemented in the tooling released with this paper.

H.1 P1: module-hook operation counting

The practice. Synaptic-operation counts are almost universally obtained by registering forward hooks on `nn.Conv1d`, `nn.Conv2d`, and `nn.Linear` modules, multiplying each layer's output shape by its kernel and fan-in, and weighting the total by a measured firing rate. This is the recipe in the widely reused counting scripts of this literature, inherited from artificial-neural-network FLOP counters, where every operation of interest does live inside a module.

Why it is wrong. A spike-driven attention operator does a large amount of arithmetic that lives in no `nn.Module` at all. In the sliding-window operator we study, the window accumulation is written as `F.pad` followed by `Tensor.unfold` and `sum`; the spiking gate is an activation; the final modulation is an elementwise product. None of these is a `Conv` or a `Linear`, so a module hook records nothing for them. The omission is not random with respect to the claim being made: this accumulation is precisely the part of the operator that a query-side router skips. A tool that cannot see it cannot measure the intervention, and the error is one-sided, because everything omitted is on the routable side.

Magnitude. On our SHD-1L configuration the uncounted window accumulation and gating amount to 0.54M operations, or 3.8% of the network's total, comparable to the entire Q projection. Because it is the most routable component, omitting it distorts the routing analysis far more than 3.8% suggests: it is 57% of everything routing can touch. A second, related error in the same script fused the QKV projection into a single routing-weighted term, when only the Q third is routable; correcting the split raises the effective cost of that projection by 63% at a balanced routing rate.

Remedy. Count tensor operations explicitly alongside module hooks. For a window of radius w over N positions at width D_e the accumulation costs $2N(2w{+}1)D_e$ operations in the `unfold` form and $\mathcal{O}(ND_e)$ with a prefix sum; gating and modulation add $2ND_e$. As a diagnostic, any operation-counting script whose total is invariant to changing w is not counting the window.

H.2 P2: TRAINING-MODE VERSUS INFERENCE-MODE ROUTING STATISTICS

The practice. Routing statistics, of the form "the gate sends $x\%$ of timesteps to the identity expert", are logged during training, averaged over training batches, and reported as the model's skip rate. Energy is then computed from that rate.

Why it is wrong. The gate is a small network containing batch normalisation and sitting behind dropout. In training mode batch normalisation uses batch statistics; in evaluation mode it uses running estimates. The gate's pre-activation distribution is therefore not the same function in the two modes, and since routing is a hard $\arg\max$ over that distribution, small shifts move a nonzero mass of timesteps across the decision boundary. The number that matters for a deployment claim is the inference-mode number on held-out data.

Magnitude. On SHD, the last training epoch reports 27% identity usage while the same checkpoint in inference mode on the test split routes 3.6%, a factor of seven. For our two-layer SSC model the last training epoch reports 0.16 and 0.38 in the two blocks, against 0.05 and 0.49 at inference. The discrepancy is large and does not have a consistent sign: one block skips far less at inference, the other more, so any correction factor derived from one block would be wrong for the other.

Remedy. Report routing statistics measured with `model.eval()` on the evaluation split, and state the mode explicitly in the caption. Where a training-time trace is shown to illustrate convergence, label it as such and do not use it in an energy computation.

H.3 P3: AN UNWEIGHTED LAYER-MEAN FIRING RATE

The practice. Energy is computed as $E = E_{\mathrm{AC}} \cdot \mathrm{SOPs} + E_{\mathrm{MAC}} \cdot \mathrm{FLOPs}_{\mathrm{in}}$, and SOPs are obtained as a mean firing rate multiplied by the network's total accumulate-operation count, with the mean taken over spiking layers, unweighted. In our own implementation this was literally `sops = mean(per_layer_rates) * total_ac_flops`.

Why it is wrong. Firing rate is a per-layer quantity and must be paired with that layer's own op count, $\mathrm{SOPs} = \sum_l r_l\, u_l\, F_l$. Taking an unweighted mean over layers and applying it to the total gives every layer's rate the same influence regardless of how much arithmetic it drives. A conditional-computation mechanism makes this failure severe, because routing changes the rate of a handful of small layers, the spiking layers inside the attention expert, while touching almost none of the network's operations. Under the unweighted mean, that local change rescales the energy of the entire model, including the MLP and encoder the router never touches. The error is invisible in a static comparison of two architectures, where it roughly cancels, and becomes first-order exactly when the intervention is confined to one operator.

Magnitude. Forcing a trained gate to skip a prescribed fraction of timesteps at inference and sweeping that fraction, the two formulas disagree sharply:

Forced skip rate	Op count	Energy, per-layer SOPs	Energy, unweighted mean
0%	—	—	—
40%	-2.9%	-5.1%	-22.3%
100%	-7.1%	-10.9%	-52.3%

The unweighted-mean formula overstates the routing effect by 4.3 to $4.8\times$. Applied to the full method rather than to a forced sweep, the same defect inflates the reported reduction by 8.0 pp on SHD-1L and 17.5 pp on SSC-2L (Table 2). It is not a constant factor and it does not cancel between architectures: the comparison between the baseline operator and its single-branch variant changes sign, from $+2.4\%$ under per-layer accounting to -10.0% under the scalar formula. A reader relying on the latter would conclude that removing the long-range branch saves energy, when measurement shows it costs energy.

Remedy. Accumulate SOPs per layer. If a single scalar rate is reported for readability, weight it by each layer's operation count, and never multiply a layer-averaged rate by a network-total op count. A quick diagnostic: force the routing rate to 100% and check that the reported energy reduction does not exceed the fraction of operations the routed component actually represents.

H.4 P4: MODEL SELECTION ON THE TEST SET

The practice. The Spiking Heidelberg Digits benchmark ships a train and a test split and no validation split. The common pattern is to alias the validation loader to the test loader, evaluate every epoch, and report the maximum.

Why it is wrong. Taking a maximum over hundreds of epochs of test accuracy is model selection on the test set. The reported figure is not an estimate of held-out performance; it is an order statistic whose expectation grows with the number of epochs and with the variance of the accuracy curve. Spiking models trained with surrogate gradients have notably noisy epoch-to-epoch accuracy, which makes the bias larger here than in most vision settings.

Magnitude. Across our ablation ladder on SHD, the gap between best-over-epochs and the mean of the final ten epochs ranges from 2.1 to 3.5 percentage points, and the gap is not constant across configurations. Under best-over-epochs the ordering of two configurations reverses relative to the leak-free estimator. Since the accuracy differences at stake in this literature are routinely below one point, a configuration-dependent bias of this size can invert a conclusion, and in our data it does. A separate instance of the same family: in our implementation the training loop accepted a test loader but never called it, so every accuracy we recorded on the three-way-split benchmarks was in fact validation accuracy. The validation-to-test gap on SSC is 1.7 to 2.5 points, and correcting it moved a baseline from 85.61% to 83.12%.

Remedy. Either carve a validation split out of the training set, for which we use 10% of train, giving $7340/816/2264$, and select on it, or report accuracy at a fixed epoch, the last or the mean of the last k, and say so. Both are free, and the second requires no retraining at all, since per-epoch accuracy is already in the training log. For benchmarks that do have official validation splits, select on validation and report on test.

Table 6: All seventeen configurations under the ceiling of Proposition 1.

Model	D	L	N	s_{attn}	s_{mlp}	ρ	$\bar{\Delta}_{40}$	$\bar{\Delta}_{100}$
SpikCommander SHD-1L	128	1	100	25.4%	48.9%	27.8%	2.83%	7.07%
SpikCommander SSC-1L	256	1	100	14.8%	77.4%	21.6%	1.28%	3.21%
SpikCommander SSC-2L	256	2	100	15.4%	80.6%	21.6%	1.33%	3.34%
SpikCommander GSC-1L	256	1	101	14.8%	77.5%	21.6%	1.28%	3.21%
SpikCommander GSC-2L	256	2	101	15.4%	80.6%	21.6%	1.33%	3.34%
SpikCommander SHD-1L (alpha=4)	128	1	100	14.7%	70.6%	27.8%	1.63%	4.08%
Spikformer-audio (SSC/GSC 2L)	256	2	100	37.2%	62.2%	58.2%	8.65%	21.63%
IML-Spikeformer (ASR enc.)	256	12	250	42.6%	57.3%	66.4%	11.32%	28.31%
Spikformer (CIFAR)	384	4	64	30.8%	56.9%	53.8%	6.64%	16.60%
Spikformer (ImageNet)	512	8	196	24.2%	40.6%	58.0%	5.61%	14.02%
Spikformer V2 (ImageNet)	512	8	196	24.2%	40.6%	58.0%	5.61%	14.02%
Spike-driven Transf. V2	512	10	196	34.6%	46.1%	50.0%	6.92%	17.30%
ST-SSA / STAtten (ImageNet)	768	8	196	28.7%	38.3%	50.0%	5.74%	14.36%
ST-SSA / STAtten (CIFAR)	512	2	64	34.4%	45.9%	50.0%	6.89%	17.21%
Addition-only SSA (CVPR'25)	384	4	64	30.8%	56.9%	53.8%	6.64%	16.60%
alpha-XNOR SSA (CVPR'25)	384	4	64	30.8%	56.9%	53.8%	6.64%	16.60%
SpArTa (2025)	384	4	64	30.8%	56.9%	53.8%	6.64%	16.60%

Table 7: Query-side against block-level attachment, all configurations.

Model	$\bar{\Delta}_{40}^{\mathrm{query}}$	$\bar{\Delta}_{40}^{\mathrm{drop}}$	ratio
SpikCommander SHD-1L	2.83%	29.75%	10.5×
SpikCommander SSC-1L	1.28%	36.90%	28.8×
SpikCommander SSC-2L	1.33%	38.39%	28.8×
SpikCommander GSC-1L	1.28%	36.57%	28.5×
SpikCommander GSC-2L	1.33%	38.02%	28.5×
SpikCommander SHD-1L (alpha=4)	1.63%	34.09%	20.9×
Spikformer-audio (SSC/GSC 2L)	8.65%	41.22%	4.8×
IML-Spikeformer (ASR enc.)	11.32%	43.33%	3.8×
Spikformer (CIFAR)	6.64%	36.22%	5.5×
Spikformer (ImageNet)	5.61%	26.68%	4.8×
Spikformer V2 (ImageNet)	5.61%	26.68%	4.8×
Spike-driven Transf. V2	6.92%	32.13%	4.6×
ST-SSA / STAtten (ImageNet)	5.74%	26.67%	4.6×
ST-SSA / STAtten (CIFAR)	6.89%	32.63%	4.7×
Addition-only SSA (CVPR'25)	6.64%	36.22%	5.5×
alpha-XNOR SSA (CVPR'25)	6.64%	36.22%	5.5×
SpArTa (2025)	6.64%	36.22%	5.5×

Table 8: Decomposition of the irreducible fraction $1 - \rho$, all configurations.

Model	ρ	K/V (locality)	shared proj.	locality's share of $1 - \rho$
SpikCommander SHD-1L	27.8%	24.1%	48.1%	33.3%
SpikCommander SSC-1L	21.6%	26.1%	52.2%	33.3%
SpikCommander SSC-2L	21.6%	26.1%	52.2%	33.3%
SpikCommander GSC-1L	21.6%	26.1%	52.2%	33.3%
SpikCommander GSC-2L	21.6%	26.1%	52.2%	33.3%
SpikCommander SHD-1L (α=4)	27.8%	24.1%	48.1%	33.3%
Spikformer-audio (SSC/GSC 2L)	58.2%	41.8%	0.0%	100.0%
IML-Spikeformer (ASR enc.)	66.4%	33.6%	0.0%	100.0%
Spikformer (CIFAR)	53.8%	46.2%	0.0%	100.0%
Spikformer (ImageNet)	58.0%	42.0%	0.0%	100.0%
Spikformer V2 (ImageNet)	58.0%	42.0%	0.0%	100.0%
Spike-driven Transf. V2	50.0%	33.3%	16.7%	66.7%
ST-SSA / STAtten (ImageNet)	50.0%	33.3%	16.7%	66.7%
ST-SSA / STAtten (CIFAR)	50.0%	33.3%	16.7%	66.7%
Addition-only SSA (CVPR'25)	53.8%	46.2%	0.0%	100.0%
α-XNOR SSA (CVPR'25)	53.8%	46.2%	0.0%	100.0%
SpArTa (2025)	53.8%	46.2%	0.0%	100.0%

Table 9: Windowed attention under both window implementations.

Model	s_{attn} (unfold)	s_{attn} (prefix)	$\bar{\Delta}_{40}$ (unfold)	$\bar{\Delta}_{40}$ (prefix)
SpikCommander SHD-1L	25.4%	22.5%	2.83%	1.39%
SpikCommander SSC-1L	14.8%	13.8%	1.28%	0.82%
SpikCommander SSC-2L	15.4%	14.4%	1.33%	0.85%
SpikCommander GSC-1L	14.8%	13.8%	1.28%	0.82%
SpikCommander GSC-2L	15.4%	14.4%	1.33%	0.85%
SpikCommander SHD-1L (alpha=4)	14.7%	12.8%	1.63%	0.79%

I Architecture Parameters, Sources, and Assumptions

I.1 Parameters and sources

Model	D	L	N	r	Source / note
Audio					
SpikCommander SHD-1L	128	1	100	2	1L-8-128, w=20, T=100 (SpikCommander Tab.; ref cfg mlp_ratio=1). SCR-MLP is a 4-stage block: measured 6.56M/block -¿ r_eff=2
SpikCommander SSC-1L	256	1	100	5	1L-16-256, α=4; SCR-MLP measured 65.5M/block -¿ r_eff=5
SpikCommander SSC-2L	256	2	100	5	2L-16-256, α=4
SpikCommander GSC-1L	256	1	101	5	1L-16-256; Mel input carries a MAC term in the first layer
SpikCommander GSC-2L	256	2	101	5	2L-16-256
SpikCommander SHD-1L (α=4)	128	1	100	5	as published (α=4 default) rather than the ref config's mlp_ratio=1
Spikformer-audio (SSC/GSC 2L)	256	2	100	4	Spikformer applied to spiking speech benchmarks; D and L inferred from the 2.57M-parameter audio baseline
IML-Spikeformer (ASR enc.)	256	12	250	4	song2026iml_spikeformer; 80-dim log-Mel, 2x stride-2 conv downsample; RepSSA_S is full softmax attention over the sequence
Vision (control)					
Spikformer (CIFAR)	384	4	64	4	Spikformer-4-384; 4-block SPS; 64 4x4 patches (Sec 4.1, Sec C.1)
Spikformer (ImageNet)	512	8	196	4	Spikformer-8-512; 196 16x16 patches (Sec 4.1)
Spikformer V2 (ImageNet)	512	8	196	4	V2-8-512; SCS stem, same D/8..D channel ladder (Table II)
Spike-driven Transf. V2	512	10	196	4	Meta-SpikeFormer 55M, stages 3-5 transformer blocks (Table 7); SDSA-3 = SN(Q(KT V)), linear O(ND2) (Eq 15)
ST-SSA / STAtten (ImageNet)	768	8	196	4	8-768 (Table 8); block-wise spatio-temporal attn with KT V ordering, O(TND2) (Eq 9)
ST-SSA / STAtten (CIFAR)	512	2	64	4	2-512 (Table 8)
Addition-only SSA (CVPR'25)	384	4	64	4	guo2025addonly; replaces the similarity op-

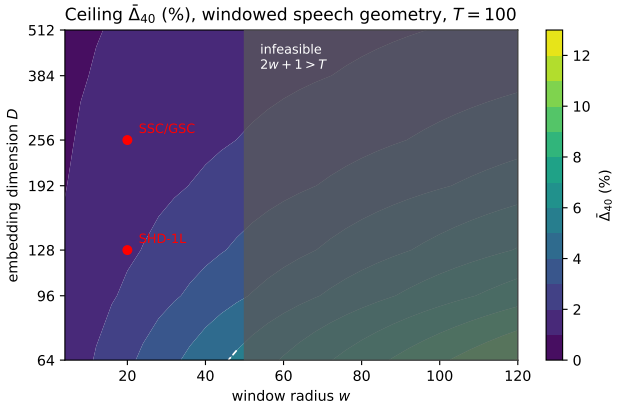

Figure 3: Ceiling at a 40% routing rate over (D, w) for windowed speech geometry. The shaded area is infeasible ($2w{+}1 > T$). Markers show published speech configurations. No feasible point exceeds 5.33% at $T = 100$.

I.2 Assumptions, and the direction each one biases the result

Every assumption below was resolved in favour of attention, i.e. against the conclusion of this paper. Adopting the opposite choice in any of them lowers s_{attn} and tightens our bound further.

- **SpikCommander SHD-1L (α=4).** The SpikCommander text gives α=4 as default while its released SHD config gives 1; we report both rows
- **Spikformer-audio (SSC/GSC 2L).** D and L inferred from the 2.57M parameter count reported for the audio baseline; full self-attention over the T=100 frame axis
- **Spikformer-audio (SSC/GSC 2L).** Costed with FULL quadratic attention (no windowing), which maximises s_attn and rho – favourable to attention
- **IML-Spikeformer (ASR enc.).** D=256, L=12 inferred from the 30.35M-parameter AiShell variant; the paper does not publish a configuration table
- **IML-Spikeformer (ASR enc.).** FFN ratio taken as 4 rather than the ASR-typical 8, which raises s_attn – favourable to attention
- **IML-Spikeformer (ASR enc.).** N=250 frames after 4x downsampling of a 10s utterance
- **Spikformer V2 (ImageNet).** MLP ratio not stated in V2; assumed 4 as in V1
- **Spike-driven Transf. V2.** Multi-stage pyramid collapsed to a single representative stage (D=512, N=196) - the stage carrying most transformer blocks
- **Spike-driven Transf. V2.** Conv stages 1-2 folded into the encoder term at 0.6x the Spikformer stem; chosen LOW so s_attn comes out HIGH (favours attention)
- **ST-SSA / STAtten (ImageNet).** MLP ratio not stated; assumed 4
- **ST-SSA / STAtten (ImageNet).** Temporal block size B folded away: it changes the constant in front of the (already linear) attention term, not its order
- **ST-SSA / STAtten (CIFAR).** MLP ratio assumed 4
- **Addition-only SSA (CVPR'25).** Backbone assumed Spikformer-4-384 (CIFAR config)
- **Addition-only SSA (CVPR'25).** Similarity op costed as full QKT - i.e. we do NOT credit their cheaper operator, which INFLATES s_attn (favours attention)
- **α-XNOR SSA (CVPR'25).** Backbone assumed Spikformer-4-384
- **α-XNOR SSA (CVPR'25).** Similarity costed at full QKT (favours attention)
- **SpArTa (2025).** Architecture not reported in the accessible text; assumed Spikformer-4-384 (its CIFAR-10 95.3% is in that range)
- **SpArTa (2025).** Costed as dense attention: crediting their sparsity would only lower s_attn further

I.3 SENSITIVITY OF THE TWO INFERRED ROWS

The two audio rows reconstructed from parameter counts are also the two least favourable to our conclusion, so we retain them and test how much their reconstruction matters. We enumerate every (D, L, r, N) combination consistent with the reported parameter count and recompute ρ and the ceiling.

Inferred row	ρ across reconstructions	$\bar{\Delta}_{40}$ across reconstructions
Spikformer-audio (2.57M)	55.8%–58.2%	8.01%–12.60%
IML-Spikeformer (30.35M)	57.0%–78.3%	5.13%–14.85%

No reconstruction yields a low ρ, so the qualitative contrast that matters — global attention retains roughly 56–78% of its operator for the router, windowed speech attention only 22–28% — does not depend on the reconstruction. Nor does any reconstruction reach the 30–49% range of reported savings.

J RECONCILING THE ATTENTION DEFINITION WITH A PER-LAYER DECOMPOSITION

Section 2 defines $C_{\mathcal{A}}$ as everything between the residual-stream taps, which is broader than the convention of counting only the similarity computation. A reader comparing our s_{attn} against a per-layer operation table will see roughly a threefold difference, and we reconcile the two here so that the discrepancy is not mistaken for an arithmetic error.

The table below is the measured per-layer decomposition of our SHD-1L configuration, obtained with the instrumented counter of §4. Total network cost is 14.00M operations per sample.

Component	Ops (M)	Share	In $C_{\mathcal{A}}$?
SEE encoder (`pwconv` + `linear`)	3.43	24.5%	no
Channel MLP (`pw1`, `fc1`, `fc2`, `pw2`)	6.56	46.9%	no
Classifier head	0.26	1.9%	no
Input projection ($D \to D_e$)	0.82	5.9%	yes, not skippable
Output projection ($D_e \to D$)	0.82	5.9%	yes, not skippable
QKV projection	1.22	8.7%	yes, 1/3 skippable
Window accumulation, gate, modulation	0.54	3.9%	yes, fully skippable
Spike-driven gate (DWConv, BN, LIF, linear)	< 0.05	$< 0.4\%$	yes, not skippable
$C_{\mathcal{A}}$ total	3.40	24.3%	—

Two observations follow.

First, the reconciliation: a reader counting only the QKV projection would report attention at 8.7%, and one counting only the similarity computation would report 3.9%. Our $s_{\text{attn}} = 25.4\%$ (the small difference from 24.3% here is that the analytic model in `attention_share.py` omits the head and the gate, which appear in the measured total) includes the two bracketing projections. The broader definition is the one that matters for this paper, because those projections are inside the boundary a router is attached to — and they are the single largest non-skippable component.

Second, this table is why ρ is small. Of the 3.40M operations in the attention sub-block, the router can eliminate the window accumulation (0.54M) and one third of the QKV projection (0.41M), for 0.95M or 27.8%. The remaining 72.2% splits into 0.82M of key/value projection that neighbouring queries consume and 1.64M of shared bracketing projections — the 1:2 ratio reported in Table 8.

Why the ratio is parameter-free. With a bottleneck $D_e = D/2$, the per-frame costs are D_e^2 for Q, $2D_e^2$ for K and V jointly, and $2 \cdot D \cdot D_e = 4D_e^2$ for the two bracketing projections. Locality thus costs $2D_e^2$ and shared projections $4D_e^2$, a ratio of exactly 1:2 independent of D, L, N, w and the MLP ratio. This is why the final column of Table 8 reads 33.3% for every SpikCommander configuration; it is an algebraic identity of the block structure, not a numerical coincidence.

K Experimental Protocol

K.1 Datasets and splits

SHD. The Spiking Heidelberg Digits (Cramer et al., 2022) contains spoken digits in English and German, encoded as spike trains over 700 input channels, in 20 classes. The benchmark ships 8156 training and 2264 test samples and *no* validation split. Following §4 we carve 10% of the training set as a validation split, giving $7340/816/2264$, and we report the mean over the final ten epochs rather than the best epoch. Both changes are needed for the accuracy numbers in this paper to be leak-free, and both are free.

SSC. Spiking Speech Commands is the spiking encoding of the Google Speech Commands corpus, with 35 classes and official train, validation and test splits. We select on validation and report on test. We note that a training loop that accepts a test loader but never calls it will silently report validation accuracy as if it were test accuracy; the gap on this dataset is 1.7 to 2.5 points.

GSC. Google Speech Commands v2 (Warden, 2018) with 35 classes and official splits, taken as log-Mel features. GSC is the only benchmark of the three whose input layer is real-valued and therefore carries a MAC term, which enters the denominator of s_{attn} and lowers the ceiling further. The GSC rows of Table 1 are analytic; we did not train on GSC, and §6 records this as a limitation.

K.2 Architecture and training

All trained models use the spiking speech transformer of Wang et al. (2026) as the base architecture: a spiking embedding extractor, L blocks each pairing a spike temporal-aware attention operator with a spiking contextual-refinement channel MLP, and a linear head. We use $T = 100$ timesteps throughout, $D = 128$ with $L = 1$ on SHD and $D = 256$ with $L = 2$ on SSC, and a sliding-window radius $w = 20$. Neurons are leaky integrate-and-fire with surrogate-gradient training (Neftci et al., 2019). Routing gates use a straight-through estimator (Bengio et al., 2013) with a hard $\arg\max$ in the forward pass and the softmax gradient in the backward pass, plus a load-balancing term at $\lambda = 0.01$.

Training runs for 500 epochs on SHD and 300 on SSC with a cosine learning-rate schedule. Every configuration in the ablation ladder shares an identical data pipeline, augmentation, schedule and estimator, so that no comparison in §3 is confounded by protocol. All runs use a single consumer GPU; a full SSC-2L run takes roughly 10 hours and an SHD run roughly 2.

K.3 Routing modes

Four routing modes beyond the learned gate were implemented for this study, all released in the supplementary material.

- **random**: i.i.d. Bernoulli routing at a fixed rate, with the gate still evaluated and charged, used for rung D.
- **spike-count**: a non-learned threshold on per-frame spike count, calibrated per batch to a target rate, used for rung E.
- **drop**: block-level dropping in the style of Mixture-of-Depths (Raposo et al., 2024). The frame leaves the block entirely, its key and value never enter neighbours' windows, and its channel MLP is masked. Used in §3.4.
- **identity-bias**: a constant added to the identity logit at inference, used to sweep the forced skip rate in §2.5 without retraining.

Because arms with stochastic routing also have stochastic *inference*, re-evaluating a saved checkpoint draws a fresh Bernoulli mask and moves accuracy by up to 0.6 points. All comparisons involving those arms therefore use the mean over the final ten epochs rather than any single evaluation pass.

K.4 Matched-rate comparison on SHD

The rate the non-learned arms were matched to is the learned gate's *training-mode* identity usage; its inference-mode usage on this dataset is 3.6%, so the comparison is not matched in the way we

Table 10: Matched-rate comparison on SHD-1L, leak-free estimator. The rate the non-learned arms were matched to is the learned gate's *training-mode* identity usage; its inference-mode usage on this dataset is 3.6%, so the comparison is not matched in the way we intended. We report it as such.

Arm	Routing rate	Acc. (%)	vs F
F learned gate	27% train / 3.6% eval	93.98	—
D random	41%	93.36	-0.62
E spike-count gate	41%	93.28	-0.70
D_{27} random	27%	93.48	-0.50
E_{27} spike-count gate	27%	92.63	-1.35

intended, and we report it as such. The matched comparison that did succeed is on SSC-2L and is reported in §3.3.

K.5 The attachment-point comparison

Following §2.6, the matched three-way comparison holds the routing *rule* fixed and varies only the attachment point, using the same i.i.d. random rule at the same rates as the block-level arm of §3.4: the attention operator alone (rung D, removing 2.7% of energy at a 41% rate), the whole block (§3.4, removing 25.9% at 40%), and the channel MLP alone. Because the block-level arm is the union of the other two, the three together decompose the saving by operator. The MLP-side mask is applied immediately before the contracting map, so that the first pointwise convolution and the convolved half of the expanding map keep running and feed the depthwise windows of neighbouring frames; masking the whole module instead would remove those taps and measure a different intervention.

K.6 Energy model

Energy follows the convention of this literature, $E = E_{\mathrm{AC}} \cdot \mathrm{SOPs} + E_{\mathrm{MAC}} \cdot \mathrm{FLOPs}_{\mathrm{in}}$ with $45\,\mathrm{nm}$ CMOS constants (Horowitz, 2014; Lemaire et al., 2023). Our two deviations from the common implementation are the subject of §4 and §4: we count tensor operations alongside module hooks, and we accumulate SOPs per layer rather than multiplying a layer-averaged firing rate by a network total. Both accountings are reported side by side in Table 2 so that our numbers can be compared with previously published ones under either convention.

L Reproducing the Tables

The supplementary material additionally contains a complete log of the modifications we made to the reference implementation, grouped into measurement fixes, new analysis tools, and new routing modes, with the effect of each on the reported numbers.