How Expressive Are Spike-Driven Transformers? An Image-Capped Finite-Trace Analysis

Anonymous authors
Paper under double-blind review

Abstract

We study the expressive power of spike-driven Transformers beyond conventional linear-region counting. Once spiking dynamics map continuous inputs to finite states, downstream Transformer layers need not create new input regions to realize different functions; instead, expressivity depends on how these states can be distinguished and recombined. We formalize this effect through *region-combination capacity*, which counts the finite interaction traces assignable to fixed LIF-induced input regions as downstream parameters vary. Our analysis reveals two fundamental bottlenecks. First, shared branches cannot assign their global state alphabets independently across tokens, imposing a state-exposure constraint. Second, multiplicative interactions can collapse distinct operand pairs into the same score. Together, these effects yield a two-budget attention envelope $D_{\mathrm{real}} \le \min\{L_{QK}, L_S\}$, separating branch-side and score-side constraints. We compose these constraints through attention, gated FFNs, residual paths, and depth to obtain a full-network capacity bound. Finite calculations test branch exposure, product distinguishability, the envelope crossover, and tightness. Static profiles diagnose the active budget, while a frozen SDT-V3 recovery study examines product collisions on real data. The resulting framework characterizes how shared branches and multiplicative interactions constrain finite-trace expressivity.

1 Introduction

Spiking neural networks (SNNs) have been widely studied for image recognition, event-based perception, and neuromorphic computing. Unlike artificial neural networks (ANNs), which communicate through high-precision activations, SNNs use sparse binary events to support event-driven computation, enabling low-latency and energy-efficient inference (Tavanaei et al., 2019; Roy et al., 2019; Rueckauer et al., 2017; Eshraghian et al., 2023). More recently, the development of SNNs has expanded beyond conventional architectures adapted from ReLU-based neural networks. Spike-driven Transformers have increasingly been explored for both vision and language modeling, extending the self-attention architecture of Vaswani et al. (2017) to sparse, event-driven computation (Zhou et al., 2023; Yao et al., 2023; 2025). These advances suggest a promising route towards energy-efficient Transformer-based vision and language models for edge and resource-constrained deployment (Ferreira et al., 2025; Tang et al., 2025; Yuan et al., 2026).

However, the theory needed to explain these models is still at an early stage. Classical ANN theory relates network width, depth, and activation geometry to function approximation and expressive power (Cybenko, 1989; Hornik, 1991; Montúfar et al., 2014; Telgarsky, 2016), whereas most existing analyses of SNN expressivity focus on temporal coding or feed-forward architectures (Maass, 1997; Maass & Markram, 2004; Mostafa, 2018; Comsa et al., 2022; Stöckl & Maass, 2021). Recent work characterizes discrete-time LIF-SNNs as piecewise-constant functions over polyhedral input regions and studies spiking self-attention through approximation and effective dimension (Nguyen et al., 2025; Guo et al., 2026). These perspectives, however, do not directly quantify the finite interaction traces jointly realizable through shared spiking branches and multiplicative attention.

Count decoding maps a T-step binary spike train to only $T+1$ scalar values. Self-attention, however, constructs s^2 query–key interactions from s tokens, suggesting that multiplicative interactions may

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.

substantially enlarge the combinatorial expressivity available to these finite-valued representations. Such an expressive gain, however, is not automatic. Because the same query and key mappings are shared across tokens, their values cannot be assigned independently to individual attention entries. Moreover, distinct and reachable operand pairs may still collapse to the same value after a dot or Hadamard product. For example, a zero query maps every key to the same raw dot-product score. This leads to the central question of this work: **how much combinatorial expressivity can a spike-driven Transformer realize from finite spike states, and what limits its growth through attention and depth?**

Linear-region analysis alone, however, becomes too coarse to characterize the expressive power of spiking Transformers. It measures expressivity through the partition of the continuous input space induced by activation boundaries. For a fixed spiking front end exposing the complete finite state to downstream layers, two inputs with the same state cannot be separated by subsequent processing (Nguyen et al., 2025). Consequently, the number of linear regions becomes insensitive to the downstream architecture, even though subsequent layers can still realize different functions by recombining the states associated with these regions. To capture this complementary source of expressivity, we define *region-combination capacity*. For fixed LIF-induced input regions, it counts the finite interaction traces that can be jointly assigned to those regions as downstream parameters vary. This shifts the analysis from counting input regions to quantifying the interaction traces realizable across the architecture. As illustrated in Figure 1, a shared branch first determines which states can coexist, multiplication then determines which operand pairs remain distinguishable, and the surviving distinctions are finally composed through attention and depth.

Our analysis identifies two distinct sources of expressive loss. Across an architecture family, a branch may realize a global alphabet of Q states, yet under any single shared parameterization only B of them may be exposed simultaneously. This *branch bottleneck* already eliminates many nominal state assignments. A second loss arises during multiplication, where distinct query–key operand pairs may collapse to the same score. At the level of a single attention head, these two constraints induce two necessary budgets. The branch budget L_{QK} bounds jointly realizable query/key assignments, while L_S bounds score-quotient assignments. Hence,

$$D^{\mathrm{real}} \le \min\{L_{QK}, L_S\}.$$

The smaller term explains why pairwise interaction does not necessarily yield pairwise growth in expressivity. In the *score-limited* regime, richer recovery can enlarge the capacity envelope by separating product collisions. In the *branch-limited* regime, enlarging the score alphabet leaves the envelope unchanged, although realized trace counts may still increase. Pair-injective recovery preserves every jointly realizable Q/K assignment in the complete score trace, leaving branch realizability as the remaining constraint. We establish the same operand-versus-product principle for gated FFNs and propagate it through value projections, residual paths, complete blocks, and depth. For homogeneous count-decoded models, this yields the full-network upper bound $\log \mathrm{TraceComb}^{(L)} = O(msLD\log(T+1))$. Our contributions are:

- **An architecture-level formulation of finite-trace expressivity.** We introduce region-combination capacity to count the finite interaction traces jointly realizable over fixed LIF-induced input regions, capturing architectural expressivity beyond conventional region counting.
- **Two structural bottlenecks and their interaction.** We identify two structural bottlenecks in SNN Transformers and show how their interaction determines the attention envelope, explaining when recovery can enlarge this envelope and when it remains constrained by shared-state exposure.
- **Full-network composition and empirical validation.** We extend the finite-trace analysis through attention, gated FFNs, residual paths, and depth to obtain a full-network expressivity bound. We further validate the predicted bottlenecks and their operating regimes through experiments.

2 Related Work

Spike-driven Transformers. Recent SNN research has extended spike-based computation from conventional architectures to Transformers, including spike-driven attention, redesigned residual

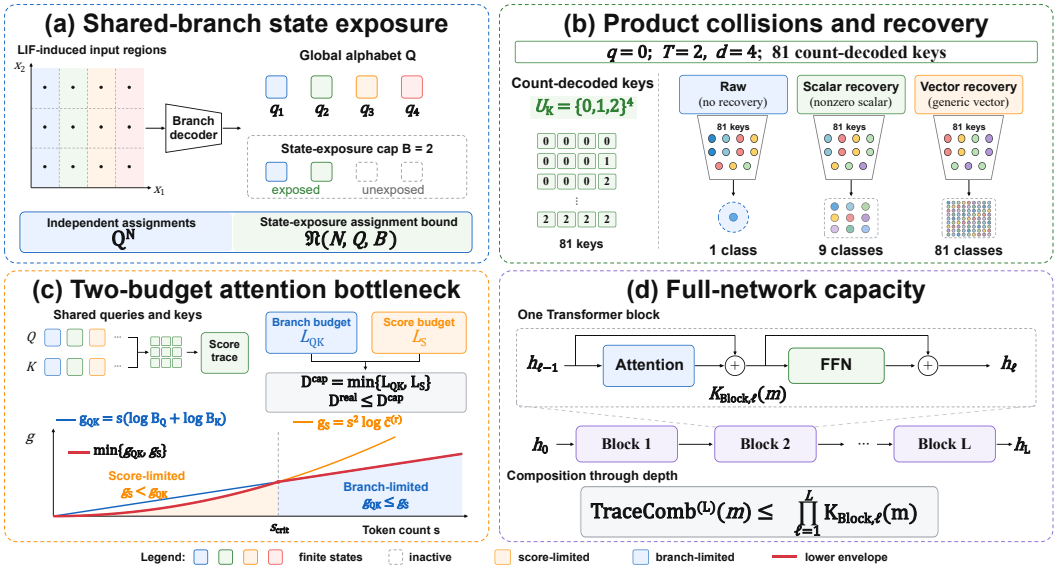


Figure 1: Overview of the proposed expressivity analysis for spike-driven Transformers.

connections, hierarchical backbones, and increasingly complex token interactions (Zhou et al., 2023; 2026; Yao et al., 2023; 2024; Shi et al., 2024; Zhou et al., 2024a;b; Yao et al., 2025). These models introduce shared query–key branches and multiplicative interactions that are absent from standard feed-forward SNNs. Existing theoretical studies have mainly focused on approximation properties or effective dimension (Guo et al., 2026), while the finite-state expressivity of these interaction mechanisms remains largely unexplored.

Linear-region expressivity. Linear-region analysis characterizes neural-network expressivity through the partition of the input space induced by nonlinear activations, with classical results establishing how width and depth affect region complexity in ReLU networks (Montúfar et al., 2014; Telgarsky, 2016; Hu & Zhang, 2018; Xiong et al., 2020; Hanin & Rolnick, 2019). This perspective has recently been extended to discrete-time LIF-SNNs, which realize piecewise-constant functions over polyhedral regions (Nguyen et al., 2025). However, once the continuous input has been converted into finite spike states, downstream layers need not create new input-space regions to realize different functions. Conventional region counting therefore becomes too coarse for spike-driven Transformers. Our work instead studies how these existing finite states remain distinguishable and are recombined through attention, gated interactions, and depth.

3 Preliminaries

Before introducing our theoretical framework, we review the basic SNN neuron model and the existing linear-region results underlying our analysis. We begin with the discrete-time leaky integrate-and-fire (LIF) neuron.

LIF dynamics and spike regions. The leaky integrate-and-fire (LIF) neuron is a standard spiking neuron model in which the membrane potential integrates incoming signals over time with leakage and emits a binary spike once a threshold is reached. For a token $x \in \mathbb{R}^n$, let $z(x) = \langle w, x \rangle + b$. A constant-drive, soft-reset LIF neuron (Gerstner et al., 2014) evolves for $t \in [T] := \{1, \dots, T\}$ as

$$s(t;x) = H(\beta u(t-1;x) + z(x) - \vartheta)\,, \qquad u(t;x) = \beta u(t-1;x) + z(x) - \vartheta s(t;x), \quad (1)$$

where $u(t;x)$ is the membrane potential, $s(t;x) \in \{0,1\}$ is the emitted spike, $\vartheta > 0$ is the firing threshold, $\beta \in (0,1]$ is the leakage factor, and $H(v) = \mathbf{1}\{v \geq 0\}$. The initial membrane potential $u(0;x) = u_0$ is independent of x.

Following common practice in modern SNNs (Rueckauer et al., 2017; Wu et al., 2019; Fang et al., 2021), we consider direct input coding, in which the same analog input is presented to the network at

each simulation step. Compared with rate or temporal encoding, direct coding avoids the additional discretization and stochasticity introduced by an input encoder, allowing the analysis to focus on the representational effect of the spiking dynamics themselves (Nguyen et al., 2025). Under this setting, Nguyen et al. (2025) established the following linear-region bound for discrete-time LIF neurons.

Lemma 1 (Single-neuron Time-to-Spike separation; Nguyen et al. (2025))**.** *Over T steps, one LIF neuron has at most*

$$\rho_T = 1 + \frac{T(T+1)}{2} \tag{2}$$

regions on which the complete spike history $(s(1;x), \dots, s(T;x))$ is constant, and this bound is attainable.

For a branch of d such neurons, let $\kappa_T = T(T+1)/2$. The corresponding branch-level region bound is

$$R(n,d,T) := \begin{cases} \sum_{i=0}^{n} \binom{d}{i} \kappa_T^i, & d \geq n, \\ \rho_T^d, & d < n. \end{cases} \tag{3}$$

Temporal decoding and recovery. Let $S \in \{0,1\}^{d \times T}$ record the spikes of d neurons over T steps, and let $a = (a_1, \dots, a_T)^\top$ be a fixed temporal decoder.

$$\Lambda_a(S) = Sa, \qquad \Sigma(a) = \left\{ \sum_{t=1}^{T} a_t \epsilon_t : \epsilon_t \in \{0,1\} \right\}, \qquad \sigma(a) = |\Sigma(a)|. \tag{4}$$

The decoded branch lies in the nominal grid $\Sigma(a)^d$, and every realized spike set $\mathcal{S}$ satisfies $|\Lambda_a(\mathcal{S})| \leq \min\{|\mathcal{S}|, \sigma(a)^d\}$. Count decoding has $\sigma(\mathbf{1}_T) = T+1$, whereas distinct-subset-sum weights preserve all 2^T binary histories. These alphabet sizes characterize the finite states retained after temporal decoding, rather than the number of input regions. In this paper, we further consider three recovery maps:

$$D_0(S) = Sa, \qquad D_z(S) = Sa - z\mathbf{1}_d, \qquad D_c(S) = Sa + \mathbf{c}. \tag{5}$$

These maps correspond to raw decoding, scalar zero-point recovery, and vector shift recovery, respectively, where $\mathbf{1}_d$ is the all-ones vector. At the theorem level, we allow $z \in \mathbb{R}$ and $c \in \mathbb{R}^d$; integer zero-points are also covered by the results in Section 4.3. The selected recovery parameters remain fixed during capacity counting.

4 Theory: From Linear Regions to Transformer Capacity

With these preliminaries in place, we now turn to our theoretical framework for characterizing the expressive capacity of spike-driven Transformers. Existing linear-region analysis describes how the spiking front end partitions the continuous input space, but this partition alone is insufficient to characterize the downstream Transformer architecture once the representation has become finite-valued. We therefore shift the object of analysis from the number of input-space regions to the finite behaviors that can be realized on these fixed regions. Specifically, we first define the corresponding region-combination capacity, then analyze how shared branch parameters and multiplicative interactions restrict the available distinctions, and finally compose these constraints through attention, gated FFNs, residual paths, and depth.

4.1 Region-combination capacity on fixed regions

Let ϕ denote the parameters of the LIF layers that induce the input-space regions characterized by the linear-region result in Lemma 1. We refer to these layers collectively as the *spiking front end*. As characterized by the linear-region result above, this front end induces a finite partition of the full-sequence input space, $\mathcal{P}_\phi = \{R_i^\phi\}_{i=1}^{m_\phi}$, where $m_\phi = |\mathcal{P}_\phi|$, such that the complete finite state exposed to downstream modules is constant within each region. For the LIF front end considered here, Equation 3 gives the region bound for each LIF branch used in this construction. When spike timing is retained and used by later modules, it is included as part of this finite state.

We further fix an observation π that records the selected finite interaction trace in a common finite alphabet $\mathcal{Z}_\pi$. For a given downstream parameter realization ψ, every region R_i^ϕ is assigned one finite trace. Collecting these traces over all regions gives one complete region-to-trace assignment.

Let $F_{\phi,\psi}$ denote the network computation with front-end parameters ϕ and downstream parameters $\psi \in \Theta_{\mathrm{down}}(\phi)$, where $\Theta_{\mathrm{down}}(\phi)$ is the admissible downstream parameter set. The integer m bounds the number of front-end regions.

Definition 1 (Region-combination capacity)**.** *For an architecture $\mathcal{A}$ and a fixed observation π, define*

$$\Gamma_{\phi,\psi}(\mathcal{P}_\phi) := \big(\pi(F_{\phi,\psi}(R_1^\phi)), \ldots, \pi(F_{\phi,\psi}(R_{m_\phi}^\phi))\big),$$
$$\mathrm{TraceComb}_m(\mathcal{A};\pi) := \sup_{\phi:\,|\mathcal{P}_\phi|\le m} \left|\{\Gamma_{\phi,\psi}(\mathcal{P}_\phi) : \psi \in \Theta_{\mathrm{down}}(\phi)\}\right|. \tag{6}$$

Here $\pi(F_{\phi,\psi}(R_i^\phi))$ denotes the common observed trace for inputs in R_i^ϕ.

For each fixed front end, $\mathrm{TraceComb}_m$ counts the region-to-trace assignments realizable by varying the downstream parameters, and then takes the supremum over admissible front ends. We begin by asking which finite states a shared branch can expose simultaneously.

4.2 State exposure under a shared branch

The first restriction arises from state coexistence. Across all parameterizations, a branch family may realize Q distinct states, whereas any single shared realization may expose only $B \le Q$ of them simultaneously. For branch b, let n_b be its input dimension, d_b its decoded width, T_b its latency, and a_b its fixed decoder. Let $U_b(\eta)$ be the set of decoded outputs obtained over the admissible branch inputs under parameters η. Define

$$Q_b^{\mathrm{nom}} := \sigma(a_b)^{d_b}, \qquad U_b^{\mathrm{glob}} := \bigcup_\eta U_b(\eta), \qquad q_b^{\mathrm{glob}} := |U_b^{\mathrm{glob}}|, \tag{7}$$

$$B_b^{\mathrm{glob}} := \sup_\eta |U_b(\eta)|, \qquad \widehat{B}_b := \min\{R(n_b, d_b, T_b),\, Q_b^{\mathrm{nom}}\}. \tag{8}$$

Here Q_b^{nom} is the nominal decoder-alphabet size, q_b^{glob} is the number of states realized somewhere in the branch family, and B_b^{glob} is the maximum number of states that can be exposed by a single realization. We refer to this simultaneous-state limit as the *branch state-exposure cap*. The bound $\widehat{B}_b$ assumes that branch b follows the constant-drive dynamics in Equation 1.

Proposition 1 (Image-capped branch assignments)**.** *Every admissible branch realization satisfies*

$$|U_b(\eta)| \le B_b^{\mathrm{glob}} \le \min\{R(n_b, d_b, T_b), q_b^{\mathrm{glob}}\} \le \widehat{B}_b. \tag{9}$$

For $N \ge 1$ fixed input slots evaluated with the same branch parameters, the number of output assignments obtained by varying those parameters is at most

$$\mathfrak{N}(N; Q, B) := \sum_{j=1}^{\min\{N,B\}} \binom{Q}{j} j!\, S(N, j), \tag{10}$$

where $(Q, B) = (q_b^{\mathrm{glob}}, B_b^{\mathrm{glob}})$ and $S(N, j)$ is the Stirling number of the second kind.

Equation 10 counts all slot assignments using at most B of Q labels, giving an upper bound for a shared branch. When the exact quantities q_b^{glob} and B_b^{glob} are unavailable, replacing them with $(Q_b^{\mathrm{nom}}, \widehat{B}_b)$ gives a computable closed-form relaxation.

Example 1 (State-exposure caps versus independent assignments). An image-capped alphabet with $Q = 3$, $B = 2$, and four shared slots has $\mathfrak{N}(4; 3, 2) = 45$ admissible assignments, compared with $3^4 = 81$ independent assignments. The missing 36 require all three states to coexist. This illustrates the state-exposure cap; it does not imply that a particular LIF branch realizes all 45 remaining assignments.

Corollary 1 (Per-slot envelope growth)**.** *For fixed integers $Q \ge B \ge 1$ and natural logarithms,*

$$\lim_{N\to\infty} \frac{1}{N} \log \mathfrak{N}(N; Q, B) = \log B. \tag{11}$$

Corollary 1 shows that B controls the asymptotic per-slot growth of the assignment envelope. When only the nominal relaxation is available, $\log \widehat{B}_b$ gives the corresponding computable upper rate. The branch-side analysis determines which finite states can be exposed simultaneously, but not whether they remain distinguishable after interaction. We next study the additional loss introduced by multiplication.

4.3 Distinction survival under multiplication

Even when two operands are available, their identities need not survive their interaction: different operand pairs may produce the same product. Let $U_q, U_k \subseteq \mathbb{R}^d$ be finite alphabets of sizes M_q, M_k, and let $C_{\cdot}^{(r)}$ and $C_{\odot}^{(r)}$ denote the numbers of distinct dot and Hadamard products under a fixed recovery rule $r \in \{0, z, c\}$ from Equation 5. For shifted recovery, we consider independently chosen scalar shifts (z_q, z_k) or vector shifts (c_q, c_k) that are generic, i.e., outside a finite union of proper affine subspaces of the corresponding joint parameter space.

Theorem 1 (Product distinguishability under recovery)**.** *For raw decoding and generic independent shifts, respectively,*

$$C_{\cdot}^{(0)} = |\{u^\top v : u \in U_q, v \in U_k\}|, \qquad C_{\cdot}^{(c)} = M_q M_k.$$
$$C_{\cdot}^{(z)} = |\{(u^\top v, \mathbf{1}^\top u, \mathbf{1}^\top v) : u \in U_q, v \in U_k\}|, \tag{12}$$

For the Hadamard channel,

$$C_{\odot}^{(0)} = |\{u \odot v : u \in U_q, v \in U_k\}|, \qquad C_{\odot}^{(z)} = C_{\odot}^{(c)} = M_q M_k.$$

Independent integer scalar zero-points outside the exceptional sets also attain the scalar-shift quantities. Consequently, $C^{(0)} \le C^{(z)} \le C^{(c)}$ for the corresponding generic choices.

For scalar dot recovery, $(u - z_q\mathbf{1})^\top (v - z_k\mathbf{1}) = u^\top v - z_k\mathbf{1}^\top u - z_q\mathbf{1}^\top v + d z_q z_k$, which explains the three invariants in Equation 12. Independent vector shifts can separate every ordered pair, whereas scalar shifts already suffice for Hadamard products. These assertions do not automatically extend to tied shifts: when both branches use the same shift, exchanging their operands still leaves a dot or Hadamard product unchanged.

For count-decoded alphabets $U_q, U_k \subseteq \{0, \dots, T\}^d$,

$$C_{\cdot}^{(0)} \le dT^2 + 1, \qquad C_{\odot}^{(0)} \le |\{ij : i, j \in \{0, \dots, T\}\}|^d. \tag{13}$$

Example 2 (A silent query exposes product collisions). Fix a zero query and all $3^4 = 81$ count-decoded keys at $T = 2$, $d = 4$. Raw dot products give one score; scalar recovery can distinguish the nine coordinate sums; generic vector recovery distinguishes all 81 keys.

Recovery can therefore separate existing operands without enlarging their branch images. Pairwise injectivity determines whether distinct operand assignments remain distinguishable; joint branch realizability determines which assignments can occur. We next combine these two aspects at the level of a complete attention head.

4.4 The two-budget attention bottleneck

We now combine the two restrictions. A complete attention head must admit both a shared operand assignment and a compatible array of product labels. Consider layer ℓ with H heads. For a fixed realization of the preceding blocks, the front-end regions induce finite states at the input of this layer. Let $m_{\ell-1}$ uniformly bound the number of distinct such states, and let s be the number of token positions associated with each state. The branch and pairwise assignments are evaluated on

$$N_{\mathrm{tok},\ell} = m_{\ell-1} s, \qquad N_{\mathrm{qk},\ell} = m_{\ell-1} s^2$$

region–token and region–token-pair slots, respectively. For head $h \in [H]$ and branch $b \in \{Q, K, V\}$, let $(q_{b,h,\ell}, B_{b,h,\ell})$ denote its alphabet size and state-exposure cap, or their nominal relaxation from Proposition 1. For fixed recovery r, let $c_{h,\ell}^{(r)}$ be the global product-alphabet size, or its nominal relaxation computed from Theorem 1, and define

$$\bar{c}_{h,\ell}^{(r)} := \min\{c_{h,\ell}^{(r)}, B_{Q,h,\ell} B_{K,h,\ell}\}. \tag{14}$$

Theorem 2 (Two-budget attention envelope). *Let $D_{h,\ell}^{\mathrm{real},(r)}$ count the jointly realizable score-quotient assignments across these regions. Then*

$$\begin{aligned} D_{h,\ell}^{\mathrm{real},(r)} &\le D_{h,\ell}^{\mathrm{cap},(r)} := \min\{L_{QK,h,\ell}, L_{S,h,\ell}\}, \\ L_{QK,h,\ell} &:= \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{Q,h,\ell}, B_{Q,h,\ell})\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{K,h,\ell}, B_{K,h,\ell}), \\ L_{S,h,\ell} &:= \mathfrak{N}(N_{\mathrm{qk},\ell}; c_{h,\ell}^{(r)}, \bar{c}_{h,\ell}^{(r)}). \end{aligned} \tag{15}$$

We call the envelope score-limited *when $L_{S,h,\ell} < L_{QK,h,\ell}$ and* branch-limited *when $L_{QK,h,\ell} \le L_{S,h,\ell}$. Including values gives the attention and multi-head upper envelopes*

$$K_{\mathrm{Attn},h,\ell} := D_{h,\ell}^{\mathrm{cap},(r)}\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{V,h,\ell}, B_{V,h,\ell}), \qquad K_{\mathrm{MHA},\ell} := \prod_{h=1}^{H} K_{\mathrm{Attn},h,\ell}, \tag{16}$$

The active budget determines which mechanism can enlarge the certified attention envelope: recovery can raise a score-limited envelope by separating product collisions, whereas increasing the score budget alone cannot raise a branch-limited envelope.

Preservation under pair-injective recovery. For fixed h, ℓ, let $\mathcal{A}_{QK}$ be the jointly realizable Q/K assignments on these region–token slots. If the fixed score map is injective on ordered operand pairs, then $D_{h,\ell}^{\mathrm{real},(r)} = |\mathcal{A}_{QK}|$ for the complete score trace. Thus recovery preserves every realizable operand assignment, without requiring independent Q/K branches (Appendix A.3).

Corollary 2 (Asymptotic envelope regimes). *For fixed s, alphabets, and positive state-exposure caps, as $m_{\ell-1} \to \infty$, the logarithmic rates per retained input state of L_{QK}, L_S, and D^{cap} are*

$$g_{QK} = s(\log B_{Q,h,\ell} + \log B_{K,h,\ell}), \qquad g_S = s^2 \log \bar{c}_{h,\ell}^{(r)}, \qquad g_{\mathrm{Attn}} = \min\{g_{QK}, g_S\}. \tag{17}$$

When the denominator is positive, their crossover occurs at $s_{\mathrm{crit}} = \frac{\log B_{Q,h,\ell} + \log B_{K,h,\ell}}{\log \bar{c}_{h,\ell}^{(r)}}$.

Here g_{Attn} concerns the score-stage envelope before including values. The crossover s_{crit} marks where the active attention envelope switches between the branch and score budgets. The attention result identifies the local bottleneck of a single head. To obtain a network-level capacity bound, we must next determine how the same distinction-survival principle propagates through the remaining Transformer components and across depth.

4.5 From local bottlenecks to full-network capacity

We now extend the local branch and product bounds to complete Transformer blocks and deep networks. We first account for the finite states propagated through the remaining components of each block, including residual paths, and then compose the resulting assignment bounds across layers. A gated FFN, including the SwiGLU family (Shazeer, 2020), has the same operand-versus-product structure. For $b \in \{\mathrm{up}, \mathrm{gate}\}$, let $U_{b,\ell}$ be its global finite operand alphabet, with $q_{b,\ell} = |U_{b,\ell}|$ and state-exposure cap $B_{b,\ell}$. For a fixed coordinate-wise gating function χ_ℓ, define

$$q_{\mathrm{FFN},\ell} := |\{u \odot \chi_\ell(v) : u \in U_{\mathrm{up},\ell}, v \in U_{\mathrm{gate},\ell}\}|, \qquad B_{\mathrm{FFN},\ell} := \min\{q_{\mathrm{FFN},\ell}, B_{\mathrm{up},\ell} B_{\mathrm{gate},\ell}\}. \tag{18}$$

The post-product assignment count has the envelope

$$K_{\mathrm{FFN},\ell} := \min\{\mathfrak{N}(N_{\mathrm{tok},\ell}; q_{\mathrm{FFN},\ell}, B_{\mathrm{FFN},\ell}), \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{\mathrm{up},\ell}, B_{\mathrm{up},\ell})\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{\mathrm{gate},\ell}, B_{\mathrm{gate},\ell})\}.$$

For a standard finite-state MLP with one retained output branch, use

$$K_{\mathrm{FFN,std},\ell} := \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{\mathrm{FFN,std},\ell}, B_{\mathrm{FFN,std},\ell}). \tag{19}$$

Here $q_{\mathrm{FFN,std},\ell}$ and $B_{\mathrm{FFN,std},\ell}$ are the alphabet size and state-exposure cap of the retained output branch. If the operands are retained separately, we use the corresponding branch-side assignment count instead of the post-product envelope.

For the network bound, π records each head's score quotient and value states, FFN product states, and any additional residual or post-spike states across layers, but not separate copies of the Q/K or up/gate operands.

Let $K_{\mathrm{res},\ell}$ and $K_{\mathrm{post},\ell}$ bound additional residual-state assignments and assignments of an optional spike-recovered output module, conditional on the preceding retained states. A fixed residual copy or fixed projection contributes 1; a trainable path that changes subsequent finite states must be counted unless its effects are already included. We write each layerwise factor as a function of m by evaluating its assignment bound on at most m retained input states; in particular, $N_{\mathrm{tok}} = ms$ and $N_{\mathrm{qk}} = ms^2$. Define

$$K_{\mathrm{Block},\ell}(m) := K_{\mathrm{res},\ell}(m) \left[\prod_{h=1}^{H} K_{\mathrm{Attn},h,\ell}(m) \right] K_{\mathrm{FFN},\ell}(m) K_{\mathrm{post},\ell}(m). \tag{20}$$

Theorem 3 (Deep SNN-Transformer capacity)**.** *Consider* $\mathrm{Trans}^{(L)} = \mathrm{Block}_L \circ \cdots \circ \mathrm{Block}_1$*, and let* m_0 *be the front-end region budget in Definition 1. Assume the stated factors uniformly bound all compatible next-layer extensions of each retained-trace prefix (Appendix A.4). If layer* ℓ *receives at most* $m_{\ell-1} \le m_0$ *distinct finite states induced by the front-end regions, then*

$$\mathrm{TraceComb}^{(L)}(m_0) \le \prod_{\ell=1}^{L} K_{\mathrm{Block},\ell}(m_{\ell-1}) \le \prod_{\ell=1}^{L} K_{\mathrm{Block},\ell}(m_0). \tag{21}$$

Here $\mathrm{TraceComb}^{(L)}$ *denotes Definition 1 for the prescribed retained-trace observation. For homogeneous layers, the last term is* $[K_{\mathrm{Block}}(m_0)]^L$*.*

Equation 21 composes the surviving finite distinctions across depth: each block extends the assignments preserved by the preceding blocks. Complete proofs are given in Appendix A.

Corollary 3 (Homogeneous count-decoded envelope)**.** *Consider homogeneous count-decoded blocks of width* D*, latency* T*, and FFN expansion ratio* γ*, with* $m = m_0$*. The total Q/K/V widths are* D *each, and the up/gate widths are* γD *each. Assume that the retained trace factors through these codes and their products, with no additional independently variable residual or post-spike states. Then*

$$\log \mathrm{TraceComb}^{(L)}(m) \le (3 + 2\gamma) msLD \log(T+1) = O(msLD \log(T+1)) \,. \tag{22}$$

This nominal-alphabet relaxation gives a simple resource-dependent envelope. An asymptotic attainment construction using persistent slot-specific tags and injective recovery is given in Appendix A.4.4.

Fixed-resource envelope comparisons. Let P denote the parameter count and U the number of neuron updates. For fixed m, s, γ and shared resource-model constants, substituting $P \propto LD^2$ or $U \propto LTD$ into Equation 22 gives the comparison scores

$$S_P = \sqrt{L} \log(T+1), \qquad S_U = \frac{\log(T+1)}{T}. \tag{23}$$

These scores compare the certified capacity envelopes under matched resource models.

5 Experiments

In this section, we evaluate the two predicted bottlenecks along four complementary steps. First, we test whether product collisions occur in learned attention states of a trained spike-driven Transformer. Second, we test class readability from recovered scores of the same frozen representation. Third, we isolate the independent restriction imposed by shared-state exposure. Finally, exact finite-model experiments test when recovery enlarges the two-budget envelope and when it instead increases realizable traces below an unchanged branch-limited bound.

Product collisions in learned attention states. We evaluate SDT-V3-S and SDT-V3-L (Yao et al., 2025), both trained from scratch on ImageNet-1K for 300 epochs and frozen during evaluation. From the same 2,000 images in a fixed 100-class ImageNet subset, we analyze post-MultiSpike Q/K count states under raw multiplication and fixed scalar/vector recovery as an offline diagnostic. For a query with $M > 1$ distinct keys, normalized distinct-key loss is $(M - C_r)/(M-1)$, where C_r is the number of distinct scores. For SDT-V3-S, across eight attention layers, raw loss is 0.646–0.901, compared with 0.290–0.537 after scalar recovery; vector recovery yields no detected collisions (Figure 2(a)). Thus, multiplication merges distinctions already present in learned Q/K states, while recovery separates them without modifying the branches. SDT-V3-L results and additional robustness checks are reported in Appendix B.6.

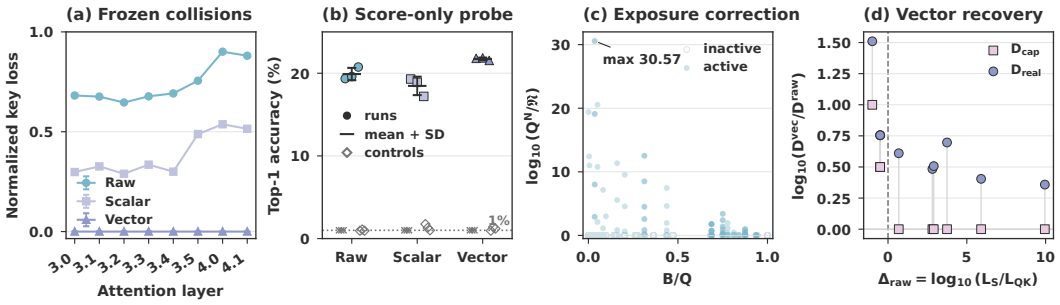


Figure 2: Evidence for product-collision and shared-state bottlenecks in learned and controlled finite models.

Do recovered distinctions remain class-readable? We train identical MLP readouts on the final-layer 16×16 score submatrices over 16 fixed spatial positions per head, using a 10,000/2,000/2,000 train/dev/test split; only the readout is trained. Test Top-1 over three runs is $(19.90 \pm 0.75)\%$, $(18.47 \pm 1.12)\%$, and $(21.73 \pm 0.16)\%$ for raw, scalar, and vector scores, respectively (mean $\pm$ sample SD; Figure 2(b)). Vector recovery improves the mean by 1.83 percentage points over raw, while constant-feature and sample-shuffle controls remain near the 1% chance level. Full probe protocols and controls are in Appendix B.6.

Shared-state restrictions beyond product collisions. Recovery separates colliding products but cannot increase how many states a shared branch exposes simultaneously. Comparing Q^N with $\mathfrak{N}(N; Q, B)$ over 960 analytic settings, the state-exposure constraint reduces the independent-slot envelope by up to 30.57 orders of magnitude (attained at $n < d$; Figure 2(c)). Hence, even collision-free scoring cannot realize state combinations that shared branches never expose together. Finite-size state-exposure and complete-alphabet recovery analyses are reported in Appendices B.1–B.2.

When does recovery enlarge expressivity? Across 240 finite configurations, recovery enlarges the envelope in 41 score-limited cases and leaves it unchanged in 199 branch-limited cases (Appendix B.3). Exact enumeration of ten capped models further shows that vector recovery can increase realizable score traces even below an unchanged branch-limited envelope (Figure 2(d)). Vector-shift dot and recovered Hadamard products attain their corresponding envelopes in the tested finite models (Appendix B.4). Thus, recovery removes product collisions locally, whereas enlargement of the attention envelope additionally depends on which budget is active.

Supplementary finite-model checks. On the 81-key alphabet $U_K = \{0, 1, 2\}^4$ with a silent query, raw, scalar-recovered, and vector-recovered dot scores retain 1, 9, and 81 classes, respectively. With recovery and readout jointly optimized, vector-dot accuracy reaches $(97.3 \pm 2.2)\%$ over five seeds on the same complete alphabet, while raw and scalar readouts reach their class-based ceilings (Appendix B.2). For independent Q/K assignments from $\{0, 1\}^2$ with $m = 1$, $s = 2$, and $B_Q = B_K = 4$, exact enumeration gives 45 raw score traces against an envelope of 81; vector recovery attains its enlarged envelope of 256 (Appendix B.4). These checks distinguish local score separation from joint trace realizability. Architecture profiles and trained-model protocols and controls appear in Appendices B.5–B.6.

6 Conclusion

We studied spike-driven Transformer expressivity from a finite-trace perspective and identified two distinct structural bottlenecks: shared branches restrict which states can be exposed jointly, while multiplicative interactions can collapse distinctions already present in those states. Their interaction yields a two-budget attention envelope with branch- and score-limited regimes, while composition through depth gives a full-network capacity bound for homogeneous count-decoded models. Recovery can increase realizable score traces without enlarging a branch-limited attention envelope. Finite-model analyses and frozen-model experiments support these distinctions, connecting the structural theory to learned representations.

REFERENCES

Iulia-Maria Comsa, Krzysztof Potempa, Luca Versari, Thomas Fischbacher, Andrea Gesmundo, and Jyrki Alakuijala. Temporal coding in spiking neural networks with alpha synaptic function: Learning with backpropagation. *IEEE Transactions on Neural Networks and Learning Systems*, 33 (10):5939–5952, 2022. doi: 10.1109/TNNLS.2021.3071976.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989. doi: 10.1007/BF02551274.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Jason K. Eshraghian, Max Ward, Emre O. Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023. doi: 10.1109/JPROC.2023.3308088.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2641–2651, 2021. doi: 10.1109/ICCV48922.2021.00266.

Pietro M. Ferreira, Siqi Wang, Yueyuan Gao, and Aziz Benlarbi-Delai. A comparative review of deep and spiking neural networks for edge AI neuromorphic circuits. *Frontiers in Neuroscience*, 19: 1676570, 2025. doi: 10.3389/fnins.2025.1676570.

Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, 2014. ISBN 9781107447615. doi: 10.1017/CBO9781107447615. URL https://doi.org/10.1017/CBO9781107447615.

Dongxin Guo, Jikun Wu, and Siu Ming Yiu. Closing the theory–practice gap in spiking transformers via effective dimension. *arXiv preprint arXiv:2604.15769*, 2026. doi: 10.48550/arXiv.2604.15769. URL https://arxiv.org/abs/2604.15769.

Boris Hanin and David Rolnick. Deep ReLU networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/9766527f2b5d3e95d4a733fcfb77bd7e-Abstract.html.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2): 251–257, 1991. doi: 10.1016/0893-6080(91)90009-T.

Qiang Hu and Hao Zhang. Nearly-tight bounds on linear regions of piecewise linear neural networks. *arXiv preprint arXiv:1810.13192*, 2018. doi: 10.48550/arXiv.1810.13192. URL https://arxiv.org/abs/1810.13192.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997. doi: 10.1016/S0893-6080(97)00011-7.

Wolfgang Maass and Henry Markram. On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences*, 69(4):593–616, 2004. doi: 10.1016/j.jcss.2004.04.001.

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://papers.nips.cc/paper_files/paper/2014/hash/fa6f2a469cc4d61a92d96e74617c3d2a-Abstract.html.

Hesham Mostafa. Supervised learning based on temporal coding in spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3227–3235, 2018. doi: 10.1109/TNNLS.2017.2726060.

Duc Anh Nguyen, Ernesto Araya, Adalbert Fono, and Gitta Kutyniok. Time to spike? understanding the representational power of spiking neural networks in discrete time. *arXiv preprint arXiv:2505.18023*, 2025. doi: 10.48550/arXiv.2505.18023. URL https://arxiv.org/abs/2505.18023.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575:607–617, 2019. doi: 10.1038/s41586-019-1677-2.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11:682, 2017. doi: 10.3389/fnins.2017.00682.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. doi: 10.48550/arXiv.2002.05202. URL https://arxiv.org/abs/2002.05202.

Xinyu Shi, Zecheng Hao, and Zhaofei Yu. Spikingresformer: Bridging resnet and vision transformer in spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5610–5619, 2024. doi: 10.1109/CVPR52733.2024.00536.

Christoph Stöckl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3(3):230–238, 2021. doi: 10.1038/s42256-021-00311-4.

Kaiwen Tang, Zhanglu Yan, and Weng-Fai Wong. Sorbet: A neuromorphic hardware-compatible transformer-based spiking language model. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 58989–59004. PMLR, 2025. URL https://proceedings.mlr.press/v267/tang25l.html.

Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural Networks*, 111:47–63, 2019. doi: 10.1016/j.neunet.2018.12.002.

Matus Telgarsky. Benefits of depth in neural networks. In *Proceedings of the 29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pp. 1517–1539. PMLR, 2016. URL https://proceedings.mlr.press/v49/telgarsky16.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1311–1318, 2019. doi: 10.1609/aaai.v33i01.33011311. URL https://ojs.aaai.org/index.php/AAAI/article/view/3929.

Huan Xiong, Lei Huang, Mengyang Yu, Li Liu, Fan Zhu, and Ling Shao. On the number of linear regions of convolutional neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10514–10523. PMLR, 2020. URL https://proceedings.mlr.press/v119/xiong20a.html.

Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2798.

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 *International Conference on Learning Representations*, 2024.

Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu, and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2973–2990, 2025. doi: 10.1109/TPAMI.2025.3530246.

Xinzhe Yuan, Xiang Peng, Bin Gu, and Huan Xiong. Plug-and-play spiking operators: Breaking the nonlinearity bottleneck in spiking transformers. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. doi: 10.48550/arXiv.2605.20289. URL https://arxiv.org/abs/2605.20289. Accepted to ICML 2026.

Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. QKFormer: Hierarchical spiking transformer using Q–K attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. doi: 10.52202/079017-0416.

Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Han Zhang, Jiaqi Wang, Huihui Zhou, Zhengyu Ma, and Yonghong Tian. Spikingformer: A key foundation model for spiking neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(3):2236–2244, 2026. doi: 10.1609/aaai.v40i3.37207. URL https://ojs.aaai.org/index.php/AAAI/article/view/37207.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *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 *Advances in Neural Information Processing Systems*, volume 37, 2024b. doi: 10.52202/079017-0322.

Ethics and Reproducibility Statement

The experiments use finite enumeration and a fixed subset of a public image benchmark. The supplementary material records the finite counting procedures, data split, recovery protocol, sampling protocol, and controls. No personally identifying data are introduced by this work.

Generative AI Usage Statement

Generative AI tools were used to assist with writing and language polishing, as well as with the implementation and debugging of experimental code used to empirically validate the theoretical results presented in this paper. The theoretical development, mathematical derivations, and scientific conclusions were conducted and verified by the authors.

APPENDIX CONTENTS AND NOTATION

Appendix contents. Appendix A gives the definitions, inherited geometric ingredients, product-quotient results, and compositional capacity proofs. Appendices B.1–B.4 quantify state-exposure corrections, finite-alphabet recovery, budget crossover, and finite-model tightness. Appendix B.5 gives static diagnostics under architecture profiles, and Appendix B.6 reports recovery in trained models and its controls.

Notation.

Symbol	Meaning
$\mathcal{A}, \phi, \psi$	Architecture, front-end parameters, and downstream parameters.
$\mathcal{P}_\phi, m_\phi, m$	Front-end atom partition, its cardinality, and a uniform atom budget with $m_\phi \le m$.
s, L, H	Token count, number of layers, and number of attention heads.
T_b, d_b, n_b	Latency, width, and input dimension of branch b.
$\pi, \Gamma_{\phi,\psi}$	Finite observation and atom-level combination signature.
$\mathrm{TraceComb}, \mathrm{OutComb}$	Retained-trace and final-output combination capacities.
$\Sigma(a_b), Q_b^{\mathrm{nom}}, q_b^{\mathrm{glob}}$	Scalar decoder image, nominal decoded-grid size, and realizable global branch-alphabet size.
$B_b^{\mathrm{glob}}, \widehat{B}_b$	Exact global simultaneous-exposure cap and its geometric closed-form relaxation.
$\mathfrak{N}(N; Q, B)$	Number of length-N assignments using at most B of Q labels.
$c_{h,\ell}^{(r)}$	Product-quotient size for head h, layer ℓ, and recovery rule r.
L_{QK}, L_S	Branch-assignment and score-assignment budgets.
$D^{\mathrm{real}}, D^{\mathrm{cap}}$	Number of realizable score-quotient assignments on the declared slots and its certified cap.
$K_{\mathrm{Attn}}, K_{\mathrm{FFN}}, K_{\mathrm{Block}}$	Attention, feed-forward, and block capacity factors.

A COMPLETE PROOFS FOR THE ARCHITECTURE-LEVEL CAPACITY BOUNDS

This appendix gives the complete arguments underlying Section 4. A front-end realization is fixed before downstream parameters vary, and Theorems 2–3 retain the post-interaction finite trace rather than the pre-product operands.

A.1 NOTATION AND REGION-COMBINATION CAPACITY

Fix an SNN-Transformer architecture $\mathcal{A}$, including its layers, heads, widths, latencies, temporal decoders, recovery rules, residual paths, and finite spike-state interfaces. Write a parameter realization as $\theta = (\phi, \psi)$, where $\phi \in \Theta_{\mathrm{front}}$ determines the front-end atomization and $\psi \in \Theta_{\mathrm{down}}(\phi)$ contains the downstream parameters. Thus

$$\mathcal{F}_{\mathcal{A}} = \{F_{\phi,\psi} : \phi \in \Theta_{\mathrm{front}},\ \psi \in \Theta_{\mathrm{down}}(\phi)\}.$$

Training is not part of the capacity definition: it selects one member of this class and cannot enlarge the architecture-level envelope.

For a fixed deterministic map $F : \Omega \to \mathbb{R}^p$, define

$$U_F := F(\Omega), \qquad M_F := |U_F|, \qquad C_F(u) := \{x \in \Omega : F(x) = u\} \quad (u \in U_F).$$

All cardinalities below are finite.

Definition A.1 (Base atoms). For a fixed front-end realization ϕ, let

$$\mathcal{P}_\phi = \{R_1^\phi, \dots, R_{m_\phi}^\phi\}$$

be its finite constant-state partition. Each atom is a region on which the selected front-end branches expose the same discrete spike state or decoded finite state to the downstream modules. For a sequence of length s, this is the joint state of all s token positions.

Definition A.2 (Finite observation or quotient). For a module with finite state space $\mathcal{S}$, let $\pi : \mathcal{S} \to \mathcal{Y}$ be the finite observation used for counting. If a module contains a real-valued deterministic computation, π denotes its induced finite spike code, finite quotient class, or explicitly specified combinatorial signature, not the unrestricted real value.

Definition A.3 (Combination signature). Conditional on ϕ, the atom-level signature of a downstream realization ψ is

$$\Gamma_{\phi,\psi}(\mathcal{P}_\phi) := \big(\pi(F_{\phi,\psi}(R_1^\phi)), \dots, \pi(F_{\phi,\psi}(R_{m_\phi}^\phi))\big) \in \mathcal{Y}^{m_\phi}.$$

Definition A.4 (Architecture-level region-combination capacity). Define

$$\mathrm{Comb}_{\mathcal{P}_\phi}(\mathcal{A};\pi) := \left|\{\Gamma_{\phi,\psi}(\mathcal{P}_\phi) : \psi \in \Theta_{\mathrm{down}}(\phi)\}\right|,$$
$$\mathrm{Comb}_m(\mathcal{A};\pi) := \sup_{\substack{\phi\in\Theta_{\mathrm{front}}:\\ |\mathcal{P}_\phi|\le m}} \mathrm{Comb}_{\mathcal{P}_\phi}(\mathcal{A};\pi).$$

For the retained interaction-trace observation used in Section 4, we write $\mathrm{TraceComb}_m(\mathcal{A};\pi) := \mathrm{Comb}_m(\mathcal{A};\pi)$.

Assumption A.1 (Post-alphabet deterministic processing). Every map applied after a finite realized alphabet, including affine maps, normalization, activations, softmax, residual projections, and MLP blocks, is deterministic in its stated finite inputs and explicit side states.

Lemma A.1 (Alphabet monotonicity for a fixed parameter). Let $F : \Omega \to \mathbb{R}^m$ have finite alphabet U_F, and let $G : \mathbb{R}^m \to \mathbb{R}^p$ be deterministic. Then

$$U_{G\circ F} = G(U_F), \qquad M_{G\circ F} = |G(U_F)| \le |U_F| = M_F.$$

With $u \sim_G u'$ iff $G(u) = G(u')$, $M_{G\circ F} = |U_F/\sim_G|$, with equality to $|U_F|$ exactly when $G|_{U_F}$ is injective. Moreover, for $y \in G(U_F)$,

$$C_{G\circ F}(y) = \bigcup_{\substack{u\in U_F\\ G(u)=y}} C_F(u).$$

Proof. By definition, every element of $U_{G\circ F}$ is $G(u)$ for some $u \in U_F$, so $U_{G\circ F} \subseteq G(U_F)$. Conversely, if $y = G(u)$ with $u \in U_F = F(\Omega)$, then $u = F(x)$ for some x and $y = G(F(x)) \in U_{G\circ F}$. Hence the two sets coincide and a deterministic map cannot increase their cardinality. The equivalence classes are precisely the nonempty fibers of $G|_{U_F}$; all fibers are singletons exactly when this restriction is injective. Finally, $G(F(x)) = y$ iff $F(x)$ belongs to one of the displayed fibers, proving the union formula. □

This lemma concerns one fixed parameter realization. It is distinct from the number of atom labelings realizable as downstream parameters vary.

A.2 Spike-Recovered Branch Budgets

Role of the geometric ingredients. The single-neuron Time-to-Spike result recalled below is due to Nguyen et al. (2025). We include the geometric ingredients to make the branch-budget instantiation self-contained. They provide bounds on the finite states exposed by a branch; the subsequent assignment analysis uses these bounds to study shared-slot labelings and Transformer interactions.

Let the latency be T and define

$$[T] := \{1, \dots, T\}, \qquad \kappa_T := \frac{T(T+1)}{2}, \qquad \rho_T := 1 + \kappa_T = \frac{T^2+T+2}{2}.$$

For $S = (s(t))_{t=1}^T \in \{0,1\}^{d\times T}$ and a fixed temporal decoder $a = (a_1, \dots, a_T)^\top$, write

$$Sa := \left(\sum_{t=1}^T a_t S_{1t}, \dots, \sum_{t=1}^T a_t S_{dt}\right)^\top, \quad \Sigma(a) := \left\{\sum_{t=1}^T a_t\epsilon_t : \epsilon_t \in \{0,1\}\right\}, \quad \sigma(a) := |\Sigma(a)|.$$

Lemma A.2 (Recalled Time-to-Spike separation; Nguyen et al. (2025)). For the constant-drive, soft-reset LIF model in Equation 1 with leakage $\beta \in (0, 1]$, one neuron observed for T steps partitions input space by parallel hyperplanes into at most

$$\rho_T = 1 + \frac{T(T+1)}{2}$$

regions, and this bound is attainable.

Proof. Write $z(x) = \langle w, x\rangle + b$. The dynamics are

$$s(t) = H(\beta u(t-1) + z(x) - \vartheta), \qquad u(t) = \beta u(t-1) + z(x) - \vartheta s(t).$$

Expanding the membrane state gives

$$u(t-1) = \beta^{t-1}u(0) + \sum_{i=0}^{t-2} \beta^i z(x) - \vartheta \sum_{i=0}^{t-2} \beta^i s(t-1-i).$$

Because $\sum_{i=0}^{t-1} \beta^i > 0$, the decision at step t can be written

$$s(t) = H(z(x) - g_{t-1}(s(1), \dots, s(t-1))),$$

where

$$g_{t-1}(a_1, \dots, a_{t-1}) = \frac{-\beta^t u(0) + \vartheta\left(1 + \sum_{i=1}^{t-1} \beta^i a_{t-i}\right)}{\sum_{i=0}^{t-1} \beta^i}.$$

Every candidate boundary therefore has normal w. A boundary indexed by a past history can add a region only inside the interval that realizes that history. Group the histories by their previous spike count $r = \sum_{j=1}^{t-1} a_j$. For fixed r, the history intervals and candidate thresholds are oppositely ordered, so at most one candidate contributes. Since $r \in \{0, \dots, t-1\}$, step t adds at most t regions. Consequently

$$N(T) \le 1 + \sum_{t=1}^{T} t = 1 + \frac{T(T+1)}{2}.$$

For tightness, take $\beta = \vartheta = 1$ and $u(0) = \alpha \in (0, 1) \setminus \mathbb{Q}$. The thresholds $(r + 1 - \alpha)/t$ lie in the corresponding previous-count intervals and are noncoincident across time, so every step adds exactly t regions. □

Definition A.5 (Time-to-Spike geometric budget). For input dimension n, decoded width d, and latency T, define

$$R(n, d, T) := \begin{cases} \sum_{i=0}^{n} \binom{d}{i} \kappa_T^i, & d \ge n, \\ \rho_T^d, & d < n. \end{cases}$$

This is the parallel-family arrangement bound: d neurons induce d parallel families with at most κ_T effective cuts each.

Lemma A.3 (Attainability of the parallel-family budget). Before temporal decoding, there are d-neuron LIF branches whose joint spike histories induce exactly $R(n, d, T)$ constant regions.

Proof. Use the tight single-neuron parameters from Lemma A.2, for which each neuron supplies κ_T distinct effective boundaries and a distinct history on every interval. Choose the d neuron normals in general position, so every subset of at most n normals is linearly independent, and choose their affine scales and offsets generically. The exceptional choices that create a coincident boundary or an intersection of more than n families form a finite union of proper algebraic sets, so they can be avoided. Adding the jth family of κ_T parallel hyperplanes then obeys the arrangement recurrence

$$A(j, n) = A(j-1, n) + \kappa_T A(j-1, n-1), \qquad A(0, n) = A(j, 0) = 1.$$

Hence $A(d, n) = \sum_{i=0}^{\min\{d,n\}} \binom{d}{i} \kappa_T^i$, which equals $R(n, d, T)$ in both cases of Definition A.5. Crossing any effective boundary changes that neuron's history, so the joint spike history is constant on, and

distinguishes, every arrangement cell. A history-injective temporal decoder preserves the same count. □

Lemma A.4 (Temporal decoder quotient). Let $\mathcal{S} \subseteq \{0,1\}^{d\times T}$ be finite and define $\Lambda_a(S) = Sa$. On $\mathcal{S}$, set $S \sim_a S'$ iff $Sa = S'a$. Then

$$|\Lambda_a(\mathcal{S})| = |\mathcal{S}/\!\sim_a| \le \min\{|\mathcal{S}|, \sigma(a)^d\}.$$

For any fixed $h \in \mathbb{R}^d$, the shifted decoder $D_h(S) = Sa + h$ induces the same quotient.

Proof. The classes of $\sim_a$ are the fibers of Λ_a. Each decoded coordinate belongs to $\Sigma(a)$, so the image is contained in $\Sigma(a)^d$ and has size at most $\sigma(a)^d$ and $|\mathcal{S}|$. Finally, $Sa + h = S'a + h$ iff $Sa = S'a$. □

Full-count interfaces and temporal conventions. The temporal decoder specifies which finite information is retained at the analyzed interface. The general results use $\sigma(a)$ explicitly. The count-decoded specialization retains all $T + 1$ count levels, without imposing a further binary decision or another deterministic coarsening of those counts. In this sense it is the full count-retaining interface within the count-based comparison family.

More generally, consider the same architecture family with fixed observations π_{fine} and π_{coarse}. If a fixed map g, independent of the varied parameters, satisfies $\pi_{\mathrm{coarse}} = g \circ \pi_{\mathrm{fine}}$, every coarse region assignment is an image of a fine assignment. Hence

$$\mathrm{TraceComb}_m(\mathcal{A}; \pi_{\mathrm{coarse}}) \le \mathrm{TraceComb}_m(\mathcal{A}; \pi_{\mathrm{fine}}).$$

This comparison concerns a declared interface and its fixed coarsenings. Timing-dependent observations that distinguish histories with the same count require the corresponding retained temporal states. They are not identified with count decoding merely because their implementations also use spikes.

Proposition A.1 (Decoded branch budget). For a constant-drive LIF branch b obeying Equation 1, with input dimension n_b, decoded dimension d_b, latency T_b, fixed decoder a_b, admissible parameters η, and decoded image $U_b(\eta)$, define

$$Q_b^{\mathrm{nom}} := \sigma(a_b)^{d_b}, \qquad U_b^{\mathrm{glob}} := \bigcup_\eta U_b(\eta), \qquad q_b^{\mathrm{glob}} := |U_b^{\mathrm{glob}}|,$$

$$B_b^{\mathrm{glob}} := \sup_\eta |U_b(\eta)|, \qquad \widehat{B}_b := \min\{R(n_b, d_b, T_b), Q_b^{\mathrm{nom}}\}.$$

Then $q_b^{\mathrm{glob}} \le Q_b^{\mathrm{nom}}$ and every realization satisfies

$$|U_b(\eta)| \le B_b^{\mathrm{glob}} \le \min\{R(n_b, d_b, T_b), q_b^{\mathrm{glob}}\} \le \widehat{B}_b.$$

Proof. The union U_b^{glob} is contained in the nominal decoder grid $\Sigma(a_b)^{d_b}$, proving the first inequality. Lemma A.2 and the parallel-family arrangement bound give at most $R(n_b, d_b, T_b)$ spike patterns to one realization, while membership in the global union gives the independent cap q_b^{glob}. Taking the supremum over η and then relaxing q_b^{glob} to Q_b^{nom} proves the chain. □

Definition A.6 (Image-capped assignment count). For a global alphabet of size Q, an image cap $B \le Q$, and N labeled slots, define

$$\mathfrak{N}(N; Q, B) := \sum_{j=1}^{\min\{N,B\}} \binom{Q}{j} j!\, S(N, j).$$

Equivalently, this is the number of maps $\tau : [N] \to [Q]$ satisfying $|\mathrm{Im}(\tau)| \le B$. Indeed, condition on image size j: choose the labels, partition the slots into j nonempty fibers, and biject those fibers with the chosen labels. Thus a branch admits the tighter count $\mathfrak{N}(N; q_b^{\mathrm{glob}}, B_b^{\mathrm{glob}})$ when its realizable union is known. Otherwise, replacing that union by the nominal decoder grid gives the certified closed-form envelope $\mathfrak{N}(N; Q_b^{\mathrm{nom}}, \widehat{B}_b)$.

What the image-capped count measures. The quantity $\mathfrak{N}(N; Q, B)$ exactly counts the class of all maps from N labeled slots to a Q-symbol alphabet whose images contain at most B labels. The

realizable labeling set of a particular parameterized branch is a subset of this class. Its additional geometric restrictions may reduce the realized count without changing the image-capped identity. Proposition 1 isolates the simultaneous-exposure restriction so that the same assignment argument can be composed across different branches and interactions.

Corollary A.1 (Per-atom growth rate). For fixed integers $Q \geq B \geq 1$,

$$\lim_{N\to\infty} \frac{1}{N} \log \mathfrak{N}(N; Q, B) = \log B.$$

Proof. For fixed j, inclusion–exclusion gives $j!S(N, j) = j^N + O((j-1)^N)$. Hence the summand with $j = B$ is $\binom{Q}{B} B^N(1 + o(1))$, while every summand with $j < B$ is $O((B-1)^N)$. Taking logarithms and dividing by N proves the claim. □

A.3 Multiplicative Quotients

Counting convention. The decoder, recovery rule, and recovery parameters are fixed within each capacity count. Product classes are defined by exact equality on the prescribed finite alphabets. A real-valued score therefore names an element of a finite product image; it is not an additional freely varying state in the counted alphabet. Recovery changes which existing operand pairs collide.

A prescribed fixed quantizer applied to this local product interface defines a further quotient. The exact distinguishability results do not assert a uniform storage-bit or noise-margin guarantee, and no such guarantee is needed for the finite-alphabet capacity defined here.

Let $U_q, U_k \subseteq \mathbb{R}^d$ be finite decoded core-value sets, $M_q = |U_q|$, $M_k = |U_k|$, and $\mathcal{P} = U_q \times U_k$. Here "generic" means outside a finite union of proper affine subspaces of the relevant parameter space. For integer zero-points, generic integer choices exist because finitely many proper affine lines cannot cover $\mathbb{Z}^2$.

A.3.1 Dot-product channel

Use the three recovery maps, with $z \in \mathbb{R}$ and $c \in \mathbb{R}^d$ at theorem level,

$$D_{\mathrm{raw}}(S) = Sa, \qquad D_{\mathrm{zp}}(S) = Sa - z\mathbf{1}_d, \qquad D_c(S) = Sa + c.$$

Theorem A.1 (Dot-product quotient). For the scalar channel $q^\top k$, define

$$\begin{aligned} C^{(0)}_{\max} &:= |\{u^\top v : u \in U_q, v \in U_k\}|, \\ C^{(z)}_{\max} &:= |\{(u^\top v, \mathbf{1}_d^\top u, \mathbf{1}_d^\top v) : u \in U_q, v \in U_k\}|, \\ C^{(c)}_{\max} &:= M_q M_k. \end{aligned}$$

For fixed zero-points or shifts, the realized alphabet sizes are at most the corresponding quantities. Generic integer scalar zero-points and generic real-valued vector shifts attain the latter two bounds, and

$$C^{(0)}_{\max} \leq C^{(z)}_{\max} \leq C^{(c)}_{\max} = M_q M_k.$$

Proof. Without shifts, two pairs collide exactly when their dot products agree, giving $C^{(0)}_{\max}$. For scalar zero-points,

$$(u - z_q\mathbf{1}_d)^\top (v - z_k\mathbf{1}_d) = u^\top v - z_k\mathbf{1}_d^\top u - z_q\mathbf{1}_d^\top v + dz_q z_k.$$

The final term is pair-independent. Pairs with the same invariant triple collide for all (z_q, z_k). If their triples differ, equality imposes one nontrivial affine equation in (z_q, z_k). Avoiding the finite collection of bad lines separates all distinct triples.

For real-valued vector shifts,

$$(u + c_q)^\top (v + c_k) = u^\top v + c_q^\top v + c_k^\top u + c_q^\top c_k.$$

For two distinct pairs, equality defines a proper affine hyperplane in $\mathbb{R}^d \times \mathbb{R}^d$. Generic shifts avoid the finite union of these hyperplanes and separate all $M_q M_k$ pairs. Nongeneric choices can only

reduce the image. Finally, projection of the zero-point invariant triple onto $u^\top v$ maps its image onto the raw image, and every triple is generated by a pair in $\mathcal{P}$, proving the hierarchy. □

Corollary A.2 (Count-decoder bottleneck for dot products). Assume $a_t = 1$ and $U_q, U_k \subseteq \{0, \dots, T\}^d$. Then

$$C^{(0)}_{\max} \le dT^2 + 1, \qquad C^{(z)}_{\max} \le \min\{M_q M_k, (dT^2+1)(dT+1)^2\}, \qquad C^{(c)}_{\max} = M_q M_k.$$

If $U_q = U_k = \{0, \dots, T\}^d$, then $M_q = M_k = (T+1)^d$, $C^{(c)}_{\max} = (T+1)^{2d}$, while $C^{(0)}_{\max} \le dT^2+1$.

Proof. Each coordinate is in $\{0, \dots, T\}$, so the integer dot product is in $\{0, \dots, dT^2\}$. The zero-point invariant triple has one coordinate with at most $dT^2 + 1$ values and two with at most $dT + 1$ values, and it also cannot exceed $M_q M_k$. The vector-shift result is Theorem A.1. □

Corollary A.3 (Silent-query bottleneck for dot products). Assume count decoding, let $u = 0$ be realized, and take $U_k \subseteq \{0, \dots, T\}^d$. A silent row distinguishes one key logit under raw recovery, at most $dT + 1$ under scalar zero-point recovery, and $|U_k|$ under generic vector-shift recovery.

Proof. The raw score is identically zero. Under scalar zero-points,

$$(-z_q \mathbf{1}_d)^\top (v - z_k \mathbf{1}_d) = -z_q \mathbf{1}_d^\top v + d z_q z_k,$$

which depends only on the total count $\mathbf{1}_d^\top v \in \{0, \dots, dT\}$. For vector shift c_q, two keys $v \neq v'$ collide only if $c_q^\top (v - v') = 0$, a proper hyperplane condition. A generic c_q avoids all finitely many such hyperplanes. □

Corollary A.4 (Tied QK symmetry bottleneck). If $U_q = U_k = U = \{u_1, \dots, u_M\}$ and query/key shifts are tied, then the score table is symmetric and

$$C_{\text{tied}} \le \frac{M(M+1)}{2}.$$

Under raw count decoding,

$$C^{(0)}_{\text{tied}} \le \min\left\{ \frac{M(M+1)}{2}, dT^2 + 1 \right\}.$$

Proof. Every tied score has the form $\widetilde{u}_i^\top \widetilde{u}_j$ and therefore equals its transpose. Ordered pairs reduce to unordered pairs with repetition; Corollary A.2 supplies the count refinement. □

A.3.2 Hadamard channel

For $u, v \in \mathbb{R}^d$, write $u \odot v = (u_1 v_1, \dots, u_d v_d)$.

Theorem A.2 (Hadamard-product quotient). Define

$$C^{(0)}_{\odot,\max} := |\{u \odot v : u \in U_q, v \in U_k\}|, \qquad C^{(z)}_{\odot,\max} = C^{(c)}_{\odot,\max} := M_q M_k.$$

Generic scalar integer zero-points and generic real-valued vector shifts attain the latter two capacities, so

$$C^{(0)}_{\odot,\max} \le C^{(z)}_{\odot,\max} = C^{(c)}_{\odot,\max} = M_q M_k.$$

Proof. The raw result is the image of $(u, v) \mapsto u \odot v$. For scalar zero-points,

$$(u - z_q \mathbf{1}_d) \odot (v - z_k \mathbf{1}_d) = u \odot v - z_q v - z_k u + z_q z_k \mathbf{1}_d.$$

For distinct pairs, choose a coordinate r on which $(u_r, v_r) \neq (u'_r, v'_r)$. Equality in that coordinate implies

$$u_r v_r - u'_r v'_r - z_q(v_r - v'_r) - z_k(u_r - u'_r) = 0,$$

a nontrivial affine condition. Avoiding the finite union of bad sets separates all pairs. The vector-shift identity

$$(u + c_q) \odot (v + c_k) = u \odot v + c_q \odot v + c_k \odot u + c_q \odot c_k$$

gives the same finite-hyperplane argument. □

Corollary A.5 (Count-decoder bottleneck for Hadamard products). For $U_q, U_k \subseteq \{0, \dots, T\}^d$, define

$$\Pi_T := \{ij : 0 \le i, j \le T,\ i, j \in \mathbb{Z}\}, \qquad \mu_T := |\Pi_T|.$$

Then

$$C^{(0)}_{\odot,\max} \le \mu_T^d \le (T^2+1)^d, \qquad C^{(z)}_{\odot,\max} = C^{(c)}_{\odot,\max} = M_q M_k$$

for generic shifts. If the full count alphabet is realized, the raw image has size exactly μ_T^d and the shifted capacities are $(T+1)^{2d}$.

Proof. Each coordinate of $u \odot v$ belongs to Π_T, so the raw image lies in Π_T^d. Under the full count alphabet, coordinates vary independently and attain every point of Π_T^d. The shifted claims follow from Theorem A.2. □

Corollary A.6 (Silent-query bottleneck is removed by Hadamard recovery). If $u = 0$ is realized, the raw Hadamard row has one output. Under scalar zero-point recovery with $z_q \neq 0$,

$$|\{(-z_q \mathbf{1}_d) \odot (v - z_k \mathbf{1}_d) : v \in U_k\}| = |U_k|,$$

and under vector-shift recovery with every coordinate of c_q nonzero,

$$|\{c_q \odot (v + c_k) : v \in U_k\}| = |U_k|.$$

Proof. The raw output is zero. The scalar expression is $-z_q v + z_q z_k \mathbf{1}_d$, which is injective in v when $z_q \neq 0$. The vector-shift expression $c_q \odot v + c_q \odot c_k$ is injective in v when all entries of c_q are nonzero. □

Corollary A.7 (Tied Hadamard symmetry bottleneck). For tied $U_q = U_k = U$ and tied shifts,

$$C_{\odot,\text{tied}} \le \frac{M(M+1)}{2}.$$

With raw count decoding,

$$C^{(0)}_{\odot,\text{tied}} \le \min\left\{ \frac{M(M+1)}{2}, \mu_T^d \right\}.$$

Proof. The table satisfies $\widetilde{u}_i \odot \widetilde{u}_j = \widetilde{u}_j \odot \widetilde{u}_i$, so only unordered pairs with repetition remain. Corollary A.5 provides the count refinement. □

A.4 Transformer Combination-Capacity Bounds

Let sequence length be s, head count be H, and suppose layer ℓ receives $m_{\ell-1}$ distinguishable finite atoms or sequence states. Set

$$N_{\text{tok},\ell} := m_{\ell-1} s, \qquad N_{\text{qk},\ell} := m_{\ell-1} s^2.$$

The cap B_b applies jointly to all slots that share branch parameters. If branches are position-specific, replace $\mathfrak{N}(m_{\ell-1}s; q_b, B_b)$ by $\prod_{t=1}^{s} \mathfrak{N}(m_{\ell-1}; q_{b,t}, B_{b,t})$.

A.4.1 Multi-head attention

For head h in layer ℓ, let $U_{Q,h,\ell}, U_{K,h,\ell}, U_{V,h,\ell}$ be finite global decoded alphabets, with cardinalities $q_{Q,h,\ell}$, $q_{K,h,\ell}$, and $q_{V,h,\ell}$, and image caps $B_{Q,h,\ell}$, $B_{K,h,\ell}$, and $B_{V,h,\ell}$. Here $B_b \le \min\{R_b, q_b\}$ for exact realizable unions under the constant-drive LIF assumptions of Proposition A.1. Under the same assumptions, an unknown union permits the conservative substitution $(q_b, B_b) = (Q_b^{\text{nom}}, \widehat{B}_b)$. For fixed recovery family $r \in \{0, z, c\}$, define the canonical finite quotient map

$$\chi^{(r)}_{h,\ell} : U_{Q,h,\ell} \times U_{K,h,\ell} \to C^{(r)}_{h,\ell}, \qquad c^{(r)}_{h,\ell} := |C^{(r)}_{h,\ell}|,$$

and the fixed-realization quotient-image cap

$$\bar{c}^{(r)}_{h,\ell} := \min\{c^{(r)}_{h,\ell}, B_{Q,h,\ell} B_{K,h,\ell}\}.$$

The labels in $C_{h,\ell}^{(r)}$ are quotient labels such as the zero-point invariant triple or the generic pair identity; they are not unrestricted real scores.

Let $D_{h,\ell}^{\mathrm{real},(r)}(m_{\ell-1}, s)$ denote the number of actually realizable cross-atom score-matrix quotient traces. Define

$$D_{h,\ell}^{\mathrm{cap},(r)}(m_{\ell-1}, s) := \min\{\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{Q,h,\ell}, B_{Q,h,\ell})\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{K,h,\ell}, B_{K,h,\ell}),\\ \mathfrak{N}(N_{\mathrm{qk},\ell}; c_{h,\ell}^{(r)}, \bar{c}_{h,\ell}^{(r)})\}.$$

Then $D_{h,\ell}^{\mathrm{real},(r)} \le D_{h,\ell}^{\mathrm{cap},(r)}$.

The retained interaction observation in the results below consists of the score quotient and subsequent finite module-interface states. It does not separately preserve Q/K operands or up/gate operands.

Theorem A.3 (Image-capped attention bound). For head h in layer ℓ,

$$K_{\mathrm{Attn},h,\ell}(m_{\ell-1}) \le D_{h,\ell}^{\mathrm{cap},(r)}(m_{\ell-1}, s)\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{V,h,\ell}, B_{V,h,\ell}).$$

Consequently,

$$K_{\mathrm{MHA},\ell}(m_{\ell-1}) \le \prod_{h=1}^{H} K_{\mathrm{Attn},h,\ell}(m_{\ell-1}).$$

Proof. The Q and K branches each fill $N_{\mathrm{tok},\ell}$ labeled slots. Definition A.6 counts their assignments under the global alphabets and fixed-realization image caps. Independently, the induced score quotient is an assignment on $N_{\mathrm{qk},\ell}$ slots with global alphabet $c_{h,\ell}^{(r)}$ and cap $\bar{c}_{h,\ell}^{(r)}$. Every realizable score trace satisfies both descriptions, so their minimum is an upper bound. The value assignment is an additional tuple component, and distinct heads are retained as separate components; multiplying their cardinality bounds proves both claims. □

Preservation under pair-injective recovery. Fix a head and layer, suppress their indices, and let $\mathcal{A}_{QK}$ be the set of jointly realizable Q/K assignments on the labeled region–token slots. For a fixed score map χ that is injective on ordered operand pairs, the map

$$F_\chi : \mathcal{A}_{QK} \longrightarrow \{\text{realizable complete score traces}\}, \qquad (Q, K) \longmapsto \big(\chi(q_{a,i}, k_{a,j})\big)_{a,i,j}$$

is a bijection. Indeed, if two complete traces agree, injectivity of χ gives equality of each corresponding operand pair. Since every query and key appears in an entry of its complete score matrix ($s \ge 1$), the two assignments agree. Surjectivity holds by the definition of the trace set. Hence $D^{\mathrm{real}} = |\mathcal{A}_{QK}|$, with no independence assumption on Q and K. If the model admits every pair of cap-respecting branch assignments independently, then $|\mathcal{A}_{QK}| = L_{QK}$; combining this with $D^{\mathrm{real}} \le D^{\mathrm{cap}} \le L_{QK}$ yields $D^{\mathrm{real}} = D^{\mathrm{cap}} = L_{QK}$. □

Corollary A.8 (Per-atom attention phase boundary). For fixed positive branch and score caps,

$$\begin{aligned} g_{QK} &:= \lim_{m_{\ell-1}\to\infty} \frac{1}{m_{\ell-1}} \log\left[\mathfrak{N}(m_{\ell-1}s; q_{Q,h,\ell}, B_{Q,h,\ell})\, \mathfrak{N}(m_{\ell-1}s; q_{K,h,\ell}, B_{K,h,\ell})\right] \\ &= s(\log B_{Q,h,\ell} + \log B_{K,h,\ell}), \\ g_S &:= \lim_{m_{\ell-1}\to\infty} \frac{1}{m_{\ell-1}} \log \mathfrak{N}(m_{\ell-1}s^2; c_{h,\ell}^{(r)}, \bar{c}_{h,\ell}^{(r)}) = s^2 \log \bar{c}_{h,\ell}^{(r)}. \end{aligned}$$

Thus $g_{\mathrm{Attn}} = \min\{g_{QK}, g_S\}$. When the denominator is positive, equality of the two rates occurs at

$$s_{\mathrm{crit}} = \frac{\log B_{Q,h,\ell} + \log B_{K,h,\ell}}{\log \bar{c}_{h,\ell}^{(r)}}.$$

Proof. Apply Corollary A.1 to each assignment count and use $N_{\mathrm{tok},\ell} = m_{\ell-1}s$ and $N_{\mathrm{qk},\ell} = m_{\ell-1}s^2$. The minimum of two positive sequences has the smaller exponential rate. Solving $g_{QK} = g_S$ gives the displayed crossover. □

Corollary A.9 (Operand-observed attention trace). If Q and K operands are retained separately, then

$$K^{QKSV}_{\mathrm{Attn},h,\ell}(m_{\ell-1}) \le \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{Q,h,\ell}, B_{Q,h,\ell})\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{K,h,\ell}, B_{K,h,\ell})\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{V,h,\ell}, B_{V,h,\ell}).$$

Proof. Distinct Q/K assignments remain distinct when the operands themselves are part of the observation; score collisions therefore do not justify the score-side truncation. Multiplication by the value assignment count gives the result. □

A.4.2 Hadamard-gated FFN/SwiGLU

We use the standard gated-FFN/SwiGLU pattern of Shazeer (2020). Consider

$$g(x) = U_{\mathrm{up}}(x) \odot \psi(U_{\mathrm{gate}}(x)),$$

where ψ is deterministic and coordinate-wise. Let $U_{\mathrm{up},\ell}$ and $U_{\mathrm{gate},\ell}$ have sizes $q_{\mathrm{up},\ell}$ and $q_{\mathrm{gate},\ell}$, and image caps $B_{\mathrm{up},\ell}$ and $B_{\mathrm{gate},\ell}$. Define

$$q_{\mathrm{FFN},\ell} := |\{u \odot \psi(v) : u \in U_{\mathrm{up},\ell}, v \in U_{\mathrm{gate},\ell}\}|,$$
$$B_{\mathrm{FFN},\ell} := \min\{q_{\mathrm{FFN},\ell}, B_{\mathrm{up},\ell} B_{\mathrm{gate},\ell}\}.$$

Theorem A.4 (Image-capped gated FFN/SwiGLU bound). The post-product FFN trace satisfies

$$K_{\mathrm{FFN},\ell}(m_{\ell-1}) \le \min\{\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{\mathrm{FFN},\ell}, B_{\mathrm{FFN},\ell}),\ \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{\mathrm{up},\ell}, B_{\mathrm{up},\ell})\, \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{\mathrm{gate},\ell}, B_{\mathrm{gate},\ell})\}.$$

Proof. The global product alphabet has size $q_{\mathrm{FFN},\ell}$. In one realization, the product image has at most $B_{\mathrm{up},\ell} B_{\mathrm{gate},\ell}$ elements and never exceeds the global alphabet, giving the first assignment count. Counting the up and gate assignments separately gives the second. Both bound the same post-product trace, so their minimum is valid. □

If up and gate operands are separately observed, one must retain the branch-separated second term rather than use product collisions to merge operand traces.

Corollary A.10 (Standard finite-state FFN). If a non-gated MLP exposes one finite output branch with alphabet $q_{\mathrm{FFN,std},\ell}$ and image cap $B_{\mathrm{FFN,std},\ell}$, then

$$K_{\mathrm{FFN,std},\ell}(m_{\ell-1}) \le \mathfrak{N}(N_{\mathrm{tok},\ell}; q_{\mathrm{FFN,std},\ell}, B_{\mathrm{FFN,std},\ell}).$$

Proof. This is Definition A.6 applied to the single retained output branch. It is also the degenerate gated construction in which the gate exposes one constant finite symbol. □

Remark A.1 (Optional post-spike state). A fixed deterministic output projection can only merge retained trace states. If the output is a fresh spike-recovered module, it contributes another image-capped factor. For example,

$$q_{\mathrm{post},\ell} = \sigma(a_{\mathrm{post},\ell})^{d_{\mathrm{post},\ell}}, \quad B_{\mathrm{post},\ell} = \min\{R(n_{\mathrm{post},\ell}, d_{\mathrm{post},\ell}, T_{\mathrm{post},\ell}), q_{\mathrm{post},\ell}\}.$$

Use $K_{\mathrm{post},\ell} = 1$ if no fresh output state is retained; otherwise use $\mathfrak{N}(N_{\mathrm{post}}; q_{\mathrm{post},\ell}, B_{\mathrm{post},\ell})$ with the slot count of the exposed interface.

A.4.3 One-layer and multi-layer bounds

Let $K_{\mathrm{res},\ell}(m)$ and $K_{\mathrm{post},\ell}(m)$ be the residual and optional post-spike assignment counts. Each is 1 when no fresh finite state is exposed, or an image-capped count with the appropriate slot number otherwise.

Theorem A.5 (One-layer image-capped block bound). Define

$$K_{\mathrm{Block},\ell}(m) := K_{\mathrm{res},\ell}(m) \left[\prod_{h=1}^{H} K_{\mathrm{Attn},h,\ell}(m)\right] K_{\mathrm{FFN},\ell}(m) K_{\mathrm{post},\ell}(m).$$

Then $\mathrm{Comb}_{\mathrm{Block},\ell}(m) \le K_{\mathrm{Block},\ell}(m)$ for the retained interaction trace.

Proof. The retained block trace is a tuple consisting of the residual side state, the score and value traces of all heads, the FFN product trace, and an optional post-spike trace. The number of tuples is at most the product of component counts. Each component already incorporates its global alphabet and fixed-realization image cap, so no extra per-atom power is introduced. □

Theorem A.6 (L-layer SNN-Transformer bound). Let $\mathrm{Trans}^{(L)} = \mathrm{Block}_L \circ \cdots \circ \mathrm{Block}_1$. For a retained-trace prefix $\tau_{<\ell}$, define

$$\mathcal{E}_\ell(\tau_{<\ell}) := \{\tau_\ell : \text{some admissible realization produces } (\tau_{<\ell}, \tau_\ell)\}.$$

Assume, uniformly over compatible realizations, that

$$|\mathcal{E}_\ell(\tau_{<\ell})| \le K_{\mathrm{Block},\ell}(m_{\ell-1}).$$

Any variable finite side state needed to make this union bound valid is part of the retained prefix. If layer ℓ sees at most $m_{\ell-1}$ distinguishable finite sequence states, then

$$\mathrm{Comb}^{(L)}(m_0) \le \prod_{\ell=1}^{L} K_{\mathrm{Block},\ell}(m_{\ell-1}).$$

Since $m_{\ell-1} \le m_0$,

$$\mathrm{Comb}^{(L)}(m_0) \le \prod_{\ell=1}^{L} K_{\mathrm{Block},\ell}(m_0),$$

and homogeneous layers give $\mathrm{Comb}^{(L)}(m_0) \le [K_{\mathrm{Block}}(m_0)]^L$.

Proof. The retained prefixes form a tree. For each node $\tau_{<\ell}$, the displayed set $\mathcal{E}_\ell(\tau_{<\ell})$ is the union of next-layer extensions over all parameter realizations compatible with that trace prefix, not the extensions of one representative parameter prefix. Hence every node has at most $K_{\mathrm{Block},\ell}(m_{\ell-1})$ children. Iterating this bound down the trace-prefix tree gives the product. A deterministic layer maps the original m_0 atoms to at most m_0 finite states, uniformly over the compatible realizations being compared, and every assignment count is nondecreasing in the slot count. This proves the coarse form and homogeneous specialization. □

Remark A.2 (Fixed-network state count is different). For one fixed parameter realization, deterministic maps on finite SNN states cannot increase the current alphabet and may satisfy $m_\ell \le m_{\ell-1}$. This does not contradict growth of $\mathrm{Comb}^{(L)}$, which counts how many atom-level signatures are realizable by the architecture class. Residual paths are treated as deterministic finite-state maps unless an explicit analog side channel is introduced.

A.4.4 Readable scaling, resource matching, and conditional attainment

Corollary A.11 (Homogeneous count-decoded scaling). Consider the homogeneous model in Corollary 3: model width D, head widths summing to D, gated-FFN width γD, common latency T, and no new retained residual or post-spike code. Then

$$\log \mathrm{Comb}^{(L)}(m) \le (3 + 2\gamma) msLD \log(T + 1).$$

Under the resource proxies $P = \Theta(LD^2)$, $F = \Theta(LTD^2)$, and $U = \Theta(LTD)$, the parameter and neuron-update substitutions give the two comparison scores in Equation 23. At fixed F, m, s, γ and proxy constants, the dense-compute substitution gives

$$S_F = \sqrt{L/T}\, \log(T + 1).$$

Proof. Count decoding gives at most $(T + 1)^d$ labels to a branch of width d. The assignment count on ms labeled atom–token slots is therefore at most $(T + 1)^{msd}$, regardless of whether the image cap is active. Across all heads, the Q, K, and V widths each sum to D, so the attention trace contributes at most $(T + 1)^{3msD}$. The up and gate branches each have width γD and contribute at

most $(T+1)^{2\gamma msD}$ together. Product quotients and deterministic post-processing cannot increase either count. Multiplying the block factor over L layers and taking logarithms proves the first display.

For fixed architecture constants, $D = \Theta(\sqrt{P/L})$ under a parameter budget, $D = \Theta(\sqrt{F/(LT)})$ under dense compute, and $D = \Theta(U/(LT))$ under neuron updates. Substitution gives respectively $S_P = \sqrt{L}\log(T+1)$, $S_F = \sqrt{L/T}\log(T+1)$, and $S_U = \log(T+1)/T$. If $E = \alpha F + \beta U$, the corresponding width is the positive solution of $\alpha D^2 + \beta D = E/(LT)$, and substitution gives the envelope comparison under this declared energy proxy. Finally, if equal-width designs have the same serial latency $\tau = \Theta(LT)$, substituting $L = \Theta(\tau/T)$ leaves the factor $\log(T+1)/T$. □

For an ordinary component c in layer ℓ, let $\mathcal{T}^{\mathrm{real}}_{c,\ell}(m)$ be its actually realizable finite trace set. With alphabet $q_{c,\ell}$, image cap $B_{c,\ell}$, and slot count $N_{c,\ell}(m)$, define

$$\mathcal{T}^{\mathrm{cap}}_{c,\ell}(m) := \{\tau : [N_{c,\ell}(m)] \to [q_{c,\ell}] : |\operatorname{Im}(\tau)| \le B_{c,\ell}\}.$$

Then

$$\mathcal{T}^{\mathrm{real}}_{c,\ell}(m) \subseteq \mathcal{T}^{\mathrm{cap}}_{c,\ell}(m), \qquad |\mathcal{T}^{\mathrm{cap}}_{c,\ell}(m)| = \mathfrak{N}(N_{c,\ell}(m); q_{c,\ell}, B_{c,\ell}).$$

For a QK score component, let $\mathcal{T}^{\mathrm{real},(r)}_{QK,h,\ell}(m)$ denote its real set of score-quotient traces and let its size be $D^{\mathrm{real},(r)}_{h,\ell}(m,s)$. The two-budget envelope gives $D^{\mathrm{real},(r)}_{h,\ell} \le D^{\mathrm{cap},(r)}_{h,\ell}$.

The real block trace set need not be a Cartesian product because components share inputs or parameters. It only satisfies

$$\mathcal{T}^{\mathrm{real}}_{\mathrm{Block},\ell}(m) \subseteq \mathcal{T}^{\mathrm{real}}_{\mathrm{res},\ell}(m) \times \prod_{h=1}^{H} \left(\mathcal{T}^{\mathrm{real},(r)}_{QK,h,\ell}(m) \times \mathcal{T}^{\mathrm{real}}_{V,h,\ell}(m)\right) \times \mathcal{T}^{\mathrm{real}}_{\mathrm{FFN},\ell}(m) \times \mathcal{T}^{\mathrm{real}}_{\mathrm{post},\ell}(m).$$

Set $K^{\mathrm{real}}_{\mathrm{Block},\ell}(m) := |\mathcal{T}^{\mathrm{real}}_{\mathrm{Block},\ell}(m)|$; then $K^{\mathrm{real}}_{\mathrm{Block},\ell}(m) \le K_{\mathrm{Block},\ell}(m)$.

Proposition A.2 (Sufficient conditions for attaining the retained-trace product envelope). Fix an m-atom front-end partition and suppose:

1. a finite residual spike path preserves a persistent tag e_i for every atom through every layer;
2. every Q, K, V, up, gate, residual, and optional post-spike branch is image-capped lookup complete, meaning every capped assignment on its slots is realizable;
3. different heads, branches, and layers have independently addressable coordinates or parameter blocks, and prescribed score traces are generated by compatible Q/K assignments; and
4. the observation retains the specified layer-wise interaction trace, or an injective finite encoding of it.

Lookup completeness here concerns all atom–token slots; the construction below realizes it using persistent slot-specific tags. Then

$$\mathrm{TraceComb}^{(L)}(m) = \prod_{\ell=1}^{L} K^{\mathrm{real}}_{\mathrm{Block},\ell}(m).$$

If all local closed-form envelopes, including the active score envelopes, are simultaneously attained, then

$$\mathrm{TraceComb}^{(L)}(m) = \prod_{\ell=1}^{L} K_{\mathrm{Block},\ell}(m).$$

Proof. Theorems A.5–A.6 give the upper bound: a retained network trace is specified by its layer-wise component traces. For the reverse inequality, choose one admissible real block trace at every layer. Lookup completeness realizes each prescribed ordinary branch assignment; score compatibility realizes each prescribed quotient trace with admissible Q/K assignments. Independent coordinates or parameter blocks prevent choices for different components and layers from interfering. Persistent tags keep every original atom addressable at every depth, so every tuple in $\prod_{\ell=1}^{L} \mathcal{T}^{\mathrm{real}}_{\mathrm{Block},\ell}(m)$ is realized.

This proves the first equality. Simultaneous equality of each real component set with its closed-form envelope yields the second. □

Corollary A.12 (Near-tight tagged count-code construction). Fix m, s, L, T and γ, and let the total model width satisfy $D > ms$. There is a count-decoded LIF Transformer with persistent finite tags, generic vector-shift recovery in attention, and injective Hadamard recovery in the gated FFN whose retained-trace capacity satisfies

$$\log \mathrm{TraceComb}^{(L)}(m) \geq (3 + 2\gamma)msL(D - ms)\log(T + 1).$$

Consequently, the ratio between its exponent and the upper exponent of Corollary A.11 is at least $1 - ms/D$ and tends to one as $D/(ms) \to \infty$. If the tags are supplied outside the counted width, the upper exponent is attained exactly.

Proof. Index the ms atom–token slots by $r = (i, t)$ and reserve a one-hot tag e_r that is copied unchanged through every residual path. Put $\widetilde{D} = D - ms$ and use the remaining coordinates for finite codes. For one LIF output coordinate, every count $0, \ldots, T$ occurs on a nonempty interval of constant scalar drives in the tight construction of Lemma A.2. Given an arbitrary desired count at each slot, a linear lookup row on the one-hot tags can therefore place the scalar drive of slot r in the corresponding interval. Repeating this independently over output coordinates realizes every map from the ms slots to $\{0, \ldots, T\}^d$ for a branch of width d.

Allocate independent lookup rows to Q, K, and V branches in every head, with head widths summing to $\widetilde{D}$. Generic vector shifts make the dot-product quotient injective on every ordered Q/K pair, so the retained attention score and value trace realizes $(T + 1)^{3ms\widetilde{D}}$ choices per layer. Give each of the up and gate branches $\gamma\widetilde{D}$ independent coordinates and use a generic injective Hadamard recovery; their retained product realizes $(T + 1)^{2\gamma ms\widetilde{D}}$ further choices. Independent parameter blocks prevent these choices from interfering, and persistent tags keep the same slots addressable after every layer. Proposition A.2 therefore multiplies the choices over L layers, giving

$$\mathrm{TraceComb}^{(L)}(m) \geq (T + 1)^{(3+2\gamma)msL\widetilde{D}}.$$

Taking logarithms and substituting $\widetilde{D} = D - ms$ proves the claim. When tags are external, $\widetilde{D} = D$. The lookup matrices use at most $O(Dms)$ parameters per branch, which is within the standard $\Theta(D^2)$ layer budget when $D \geq ms$; the parameter, dense-compute, and update scalings of Corollary A.11 are therefore preserved. □

Corollary A.13 (A fixed output readout is a trace quotient). For each fixed front end ϕ, suppose that the final output assignment is a fixed function of the retained trace assignment, independent of the varied downstream parameters. Then

$$\mathrm{OutComb}^{(L)}(m) \leq \mathrm{TraceComb}^{(L)}(m).$$

For a fixed front end, equality holds exactly when the induced map on the realizable trace-assignment set is injective.

Proof. For a fixed ϕ, let $\mathcal{T}_\phi$ be the realizable retained trace-assignment set and write the output-assignment set as $G_\phi(\mathcal{T}_\phi)$, where G_ϕ is independent of the varied parameters. Its cardinality is at most $|\mathcal{T}_\phi|$, with equality exactly when G_ϕ is injective on this set. Taking the supremum over admissible front ends proves the inequality. □

Remark A.3 (Decoder and recovery choices propagate through depth). For each branch,

$$Q_b^{\mathrm{nom}} = \sigma(a_b)^{d_b}, \qquad \widehat{B}_b = \min\{R(n_b, d_b, T_b), Q_b^{\mathrm{nom}}\}.$$

Count decoding gives $\sigma(a_b) = T_b + 1$ and $Q_b^{\mathrm{nom}} = (T_b + 1)^{d_b}$; a temporal decoder with all subset sums distinct gives $\sigma(a_b) = 2^{T_b}$ and $Q_b^{\mathrm{nom}} = 2^{d_b T_b}$. These nominal quantities yield the closed-form envelope; an exact realizable union may be smaller. Recovery changes the finite score quotient alphabet $c_{h,\ell}^{(r)}$ and its assignment term

$$\mathfrak{N}(N_{\mathrm{qk},\ell}; c_{h,\ell}^{(r)}, \bar{c}_{h,\ell}^{(r)}), \quad \bar{c}_{h,\ell}^{(r)} = \min\{c_{h,\ell}^{(r)}, B_{Q,h,\ell}B_{K,h,\ell}\}.$$

These factors propagate through

$$\mathrm{Comb}_r^{(L)}(m_0) \le \prod_{\ell=1}^{L} K^{(r)}_{\mathrm{Block},\ell}(m_{\ell-1}).$$

The layer product amplifies local decoder and quotient differences, but only through image-capped assignment counts, not an unconstrained per-atom power.

Under full count alphabets, $q_Q = q_K = (T+1)^d$. Raw dot-product attention has $c^{(0)}_{h,\ell} \le dT^2 + 1$, while generic vector-shift recovery has $c^{(c)}_{h,\ell} = (T+1)^{2d}$. Raw Hadamard attention has $c^{(0)}_{\odot,h,\ell} \le \mu_T^d$, whereas generic scalar zero-point and vector-shift recovery attain $q_Q q_K$. In every case, the replacement enters the same image-capped score term and then the same block product.

B Numerical Validation Details

This appendix quantifies the state-exposure correction, tests product recovery on a complete finite alphabet, and examines the crossover and tightness of the two-budget envelope. Static architecture profiles and trained-model experiments connect these finite counts to larger attention systems. All logarithms in this appendix use base 10.

B.1 Magnitude of the state-exposure correction

We quantify how much the state-exposure constraint tightens the independent-slot count Q^N. Here, N denotes the number of fixed input slots evaluated under the same branch parameters. For count decoding, a width-d branch with latency T has nominal alphabet size $Q = (T+1)^d$; for the distinct-subset-sum decoder, $Q = 2^{Td}$. We instantiate $B = \min\{R(n,d,T), Q\}$ using the inherited LIF region bound, giving a computable upper bound on the number of jointly exposable states. We compare Q^N with the image-capped assignment envelope $\mathfrak{N}(N;Q,B)$ over $n \in \{1,2,3\}$, $d \in \{1,2,4,8\}$, $T \in \{1,2,3,4\}$, both decoders, and $N \in \{2,3,4,5,6,8,12,16,24,32\}$, for 960 finite configurations.

The correction is active exactly when $B < \min\{N, Q\}$, which occurs in 107 of the 960 configurations; the remaining 853 retain Q^N. Among these active cases, $\log_{10}(Q^N / \mathfrak{N}(N;Q,B))$ has median 0.80 (approximately $6.3\times$) and maximum 30.57. The maximum occurs at $(n,d,T,N) = (1,8,1,32)$, where $Q = 256$, $B = 9$, $\log_{10} Q^N = 77.064$, and $\log_{10} \mathfrak{N}(N;Q,B) = 46.494$. At $T = 1$, the two decoders coincide, so both attain the same maximum (Figure 2(c)).

These results quantify the finite-size reduction induced by the state-exposure cap in Proposition 1. They are consistent with Corollary 1, which shows that, for fixed Q and B, the asymptotic per-slot growth rate of the assignment envelope is $\log B$ rather than $\log Q$. These calculations quantify the state-exposure correction and are not intended as an estimate of the exact capacity of a particular LIF parameterization.

B.2 Recovery on a complete finite alphabet

We isolate how product collisions limit key identification using the constructed alphabet $U_K = \{0,1,2\}^4$, with $T = 2$, $d = 4$, and $|U_K| = 81$, and a silent query $q = 0$. This construction does not assume that a particular LIF branch jointly realizes the entire alphabet. For fixed recovery maps, raw dot scores collapse all keys to one class, scalar recovery distinguishes the nine coordinate sums, and generic vector recovery distinguishes all 81 keys. Hadamard products yield one raw class and 81 classes under either recovery. Enumeration over the complete alphabet gives the class counts in Table 1, consistent with Theorem 1. For 81 equiprobable keys and C fixed score classes, score-only identification accuracy is at most $C/81$.

To test whether these distinctions are accessible to a common readout, we use a $4 \to 64 \to 81$ ReLU MLP in every condition, repeating a scalar dot score across four input coordinates. All 81 keys form one training batch; accuracy is evaluated on this same complete alphabet. We train for 800 epochs with Adam (Kingma & Ba, 2015), learning rate 10^{-2}, and five seeds, and report final-epoch accuracy. Scores are not normalized. Scalar and vector shifts are initialized from $\mathcal{N}(0, 0.1^2)$ and optimized

jointly with the MLP. This readout optimization selects a recovery map and classifier; it does not redefine the quotient associated with a fixed map.

Table 1: Fixed-map product classes and key-identification accuracy on the complete 81-key alphabet. Readout accuracy is the mean ± sample SD over five seeds; zero SDs are omitted. The ceiling $C/81$ concerns a fixed score map, whereas the readout jointly optimizes recovery and classifier parameters.

Operator	Recovery	Classes C	Ceiling $C/81$	Readout accuracy
Dot	Raw	1	0.012	0.012
Dot	Scalar	9	0.111	0.111
Dot	Vector	81	1.000	0.973 ± 0.022
Hadamard	Raw	1	0.012	0.012
Hadamard	Scalar	81	1.000	1.000
Hadamard	Vector	81	1.000	1.000

Raw and scalar dot readouts reach their class-based ceilings, while vector dot recovery achieves 0.973 ± 0.022; both recovered Hadamard readouts reach 1.000. Across three control seeds, evaluation-time score-row shuffling yields accuracy between 0.000 and 0.012, and replacing scores by their mean yields 0.012, close to chance $1/81$. These results connect surviving product classes to key identification under the stated readout, without asserting generalization to unseen keys.

B.3 Crossover between the branch and score budgets

We study when recovery enlarges the attention envelope in an abstract capped finite assignment model. Both branches use $U_Q = U_K = \{0,1\}^2 = \{(0,0),(0,1),(1,0),(1,1)\}$, and any slot assignment using at most B_Q or B_K distinct states is admitted on its respective branch. This model specifies admissible assignments directly, independently of a concrete LIF realization. Let m be the number of retained front-end states and s the token count, so that $N_{\mathrm{tok}} = ms$ and $N_{\mathrm{qk}} = ms^2$. Pair enumeration gives dot-product quotient sizes $c_{\mathrm{raw}} = 3$, $c_{\mathrm{scalar}} = 10$, and $c_{\mathrm{vector}} = 16$ for generic fixed recovery maps. The budgets are

$$L_{QK} = \mathfrak{N}(ms; 4, B_Q)\ \mathfrak{N}(ms; 4, B_K),$$
$$L_S^{(r)} = \mathfrak{N}\big(ms^2; c_r, \min\{c_r, B_Q B_K\}\big), \qquad D_{\mathrm{cap}}^{(r)} = \min\{L_{QK}, L_S^{(r)}\}.$$

The sweep crosses $m \in \{1,2,4\}$, $s \in \{2,\dots,6\}$, and $B_Q, B_K \in \{1,\dots,4\}$, giving 240 configurations. We locate each configuration relative to the crossover using

$$\Delta \equiv \Delta_{\mathrm{raw}} = \log_{10}(L_S^{\mathrm{raw}}/L_{QK})\,.$$

There are 41 score-limited configurations ($\Delta < 0$) and 199 branch-limited configurations ($\Delta > 0$). For recovery r, the envelope gain in Figure 3(a) is $G_{\mathrm{cap}}^{(r)} = \log_{10}(D_{\mathrm{cap}}^{(r)}/D_{\mathrm{cap}}^{\mathrm{raw}})$. Both scalar and vector recovery enlarge the envelope in all 41 score-limited cases and leave it unchanged in all 199 branch-limited cases.

We also exactly enumerate ten tractable capped finite models. For each admissible pair of Q/K assignments, we form the complete ms^2-entry score-class trace; $D_{\mathrm{real}}^{(r)}$ counts the distinct complete traces, rather than distinct individual scores. The corresponding realized gain is $G_{\mathrm{real}}^{(r)} = \log_{10}(D_{\mathrm{real}}^{(r)}/D_{\mathrm{real}}^{\mathrm{raw}})$. Figure 3(b) shows positive realized gains also in branch-limited cases: recovery can increase D_{real} while D_{cap} remains unchanged. The active-budget crossover therefore determines sensitivity of the envelope, while compatibility among scores sharing Q/K operands can still limit its attainment.

B.4 Tightness of the two-budget envelope

To measure how much compatibility among shared operands restricts attainment of the envelope, we use the same four-state alphabet as in Appendix B.3. We compare dot and Hadamard products, raw, scalar, and vector recovery, and independent versus tied Q/K assignments over ten tractable settings, giving 120 exactly enumerated configurations. Tied means that Q and K use the same slot assignment, with shared cap $\min\{B_Q, B_K\}$; it does not mean that their recovery shifts are tied. Its

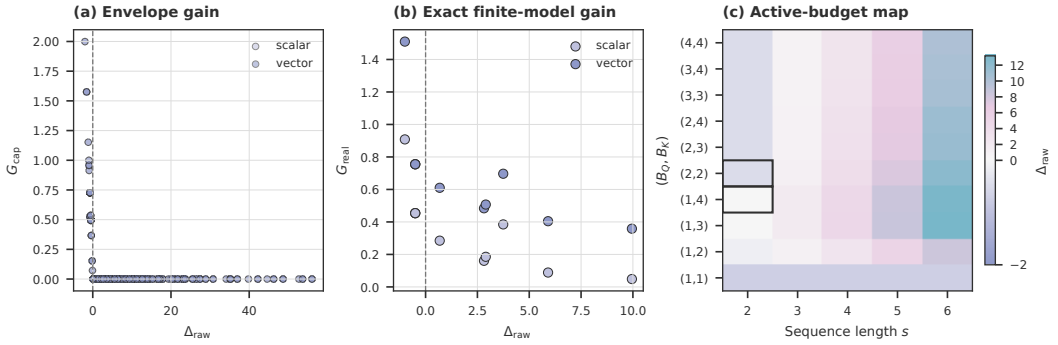


Figure 3: Two-budget crossover for independent assignments on the four-state alphabet. (a) Envelope gains G_{cap} over 240 configurations. (b) Realized gains G_{real} in ten exactly enumerated finite models. (c) The raw budget gap Δ_{raw} at $m = 1$ for the ten cap pairs with $B_Q \le B_K$; negative values are score-limited and positive values are branch-limited. Scalar and vector denote the two recovery rules.

matched branch budget is therefore $\mathfrak{N}(ms; 4, \min\{B_Q, B_K\})$, whereas independent assignments use the two-branch budget L_{QK}. We report $\tau_{\log} = \log D_{\mathrm{real}} / \log D_{\mathrm{cap}}$ only for $D_{\mathrm{cap}} > 1$.

For independent assignments, median $\tau_{\log}$ is 0.855 for raw dot, 0.915 for scalar dot, and 0.935 for raw Hadamard (Table 2). Vector dot and both recovered Hadamard channels attain their respective envelopes in every tested configuration. For these pair-injective channels, the preservation identity in Appendix A.3 explains the equality: independent capped assignments exhaust the branch budget, so $D_{\mathrm{real}} = |\mathcal{A}_{QK}| = L_{QK} = D_{\mathrm{cap}}$. Tied raw Hadamard also attains its matched envelope, with median 1.000. These ratios compare tightness against the appropriate budget in each condition; they do not imply that tying increases capacity. The tied raw-Hadamard equality is an enumeration result for the tested configurations; the pair-injective identity does not apply to that channel.

Table 2: Median logarithmic tightness over ten finite settings per condition. Independent and tied Q/K are compared with their respective two-branch and one-branch envelopes. Every condition with median 1.000 also has minimum and maximum 1.000 across the tested settings.

Operator	Recovery	Independent Q/K	Tied Q/K
Dot	Raw	0.855	0.870
Dot	Scalar	0.915	0.870
Dot	Vector	1.000	1.000
Hadamard	Raw	0.935	1.000
Hadamard	Scalar	1.000	1.000
Hadamard	Vector	1.000	1.000

A concrete incompatibility occurs at $m = 1$, $s = 2$, and $|U_Q| = |U_K| = B_Q = B_K = 4$. Independent assignments give $L_{QK} = 256$, while raw dot scores give $L_S = 81$ and $D_{\mathrm{cap}} = 81$; enumeration yields only $D_{\mathrm{real}} = 45$ complete score matrices. For example, $\left[\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ cannot equal $QK^\top$ for binary two-dimensional row operands. Its top-left entry forces $q_1 = k_1 = (1, 1)$; the zero off-diagonal entries then force $q_2 = k_2 = 0$, contradicting the bottom-right entry. Under vector recovery, the pair-injective quotient instead gives $D_{\mathrm{real}} = D_{\mathrm{cap}} = 256$ for these independent assignments. Thus a valid envelope need not be attained by a raw product channel, even though recovery can attain it in the same finite model.

B.5 Static budget regimes under declared architecture profiles

We evaluate the active term of the declared full-count envelope when its dimensional parameters are matched to SDT-V3-S and SDT-V3-L (Yao et al., 2025). These are static architecture profiles, rather than measurements of learned state occupancy. Each profile has six stage-3 and two stage-4 attention layers, with eight heads. Stage widths are $128/192$ for S and $256/360$ for L, giving head widths $d = 16/24$ and $d = 32/45$, respectively. We use the full-count interface at $T = 4$, so each Q/K

head has nominal alphabet $Q = (T+1)^d = 5^d$, independent Q/K assignments, and the geometric cap $B = \min\{R(D, d, 4), Q\}$ at stage input width D. The raw-dot score alphabet is relaxed to $c_{\mathrm{raw}} = 16d + 1$, whereas generic vector recovery has $c_{\mathrm{vector}} = Q^2$.

The retained-state budget m sets $N_{\mathrm{tok}} = ms$ and $N_{\mathrm{qk}} = ms^2$ as in Appendix B.3. At the standard sequence length $s = 196$, all eight attention layers in each profile are branch-limited, and vector recovery leaves their envelopes unchanged. This conclusion holds throughout $m \in \{1, 2, 4, 8, 16, 64, 256, 1000\}$. The Q/K exposure caps are inactive in these profiles, so $\log_{10} L_{QK} = 2ms\, d \log_{10} 5$ and the branch-limited $\log_{10} D_{\mathrm{cap}}/m$ is independent of m. The absence of a further image-cap reduction still leaves the shared-operand restriction: Q/K states occupy ms token slots, rather than ms^2 independently assignable score slots.

At $m = 1000$, shortening the L profile to $s = 16$ or $s = 8$ makes all eight layers score-limited and recovery-sensitive (Table 3). A layer is called recovery-sensitive when $\log_{10} D_{\mathrm{cap}}^{\mathrm{vector}} - \log_{10} D_{\mathrm{cap}}^{\mathrm{raw}} > 10^{-3}$ for its head profile. The switch follows the linear token dependence of the branch budget and the quadratic token-pair dependence of the score budget. The shorter sequences are evaluations of the same declared dimensions, not results from separately trained models.

Table 3: Active terms of the declared full-count envelope for SDT-V3 dimensional profiles with dot scores, $T = 4$, and $m = 1000$. Counts are out of eight attention layers. Branch- and score-limited refer to the raw-score envelope; recovery-sensitive denotes an envelope increase under vector recovery.

Profile	Tokens s	Branch-limited	Score-limited	Recovery-sensitive
SDT-V3-S	196	8	0	0
SDT-V3-L	196	8	0	0
SDT-V3-L	16	0	8	8
SDT-V3-L	8	0	8	8

B.6 Recovery in trained spike-driven Transformers

Models and data. We evaluate SDT-V3-S and SDT-V3-L (Yao et al., 2025), trained from scratch on ImageNet-1K for 300 epochs, using the final trained model of each size. The models have approximately 5.1M and 19.0M parameters, respectively, with six stage-3 and two stage-4 attention layers and eight heads per layer. Their stage-3/stage-4 head dimensions are $16/24$ for S and $32/45$ for L. Model parameters and batch-normalization statistics remain fixed throughout the analysis.

Both models use the same fixed 100-class ImageNet subset (Deng et al., 2009), with 10,000 training, 2,000 development, and 2,000 test images for the probe. Probe training and development images are disjoint subsets of the official training split; test images come from the official validation split. We extract post-MultiSpike Q/K states from every attention layer and undo the fixed output scale of $1/4$ to obtain integer counts in $\{0, \ldots, 4\}$.

Fixed recovery maps. For a query q and key k, we compare

$$s_{\mathrm{raw}}(q, k) = q^\top k, \tag{24}$$

$$s_{\mathrm{scalar}}(q, k) = (q - \alpha_q \mathbf{1})^\top (k - \alpha_k \mathbf{1}), \tag{25}$$

$$s_{\mathrm{vector}}(q, k) = (q + c_q)^\top (k + c_k). \tag{26}$$

For each layer and head, Q and K shifts are independently sampled and then fixed across images and positions. We use three recovery seeds. Scalar shifts follow $\alpha \sim \mathrm{Unif}[-2, 2]$; vector directions are normalized standard-normal draws with $\|c\|_2 = \sqrt{d}\,|\alpha|$, matching the scalar shift norm at head dimension d. Recovery parameters are not optimized. Expanded score expressions are evaluated in FP64 and grouped by exact equality, without additional rounding. Raw dot products and canonical grouping statistics use integer arithmetic. Computed scalar and vector class counts agree with their canonical counterparts in all evaluated rows.

Collision measurement. We fix eight query positions per head and use the same positions across images, layers, models, and recovery conditions. Each query is compared against all 196 native keys,

deduplicated by equality of the complete integer key vector. For a row with $M > 1$ distinct keys and C_r distinct scores under recovery r, normalized distinct-key loss is

$$\ell_r = \frac{M - C_r}{M - 1}. \tag{27}$$

Excluding rows with only one distinct key leaves 1,023,872 valid S rows and 1,024,000 valid L rows. We average valid head–query rows within each image, average the three fixed shifts for scalar and vector recovery, and then average across the 2,000 test images. Confidence intervals use 2,000 paired image-bootstrap replicates. We also repeat the calculation after excluding queries whose original count vectors are identically zero.

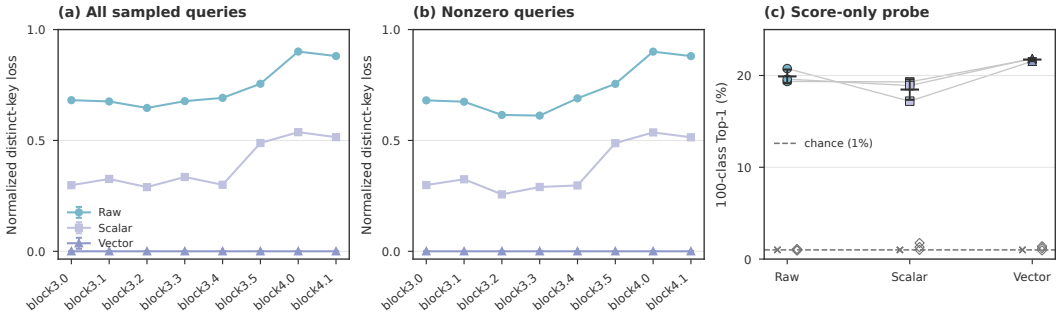


Figure 4: Recovery in frozen SDT-V3-S attention states. (a,b) Collision loss with all sampled queries and with nonzero queries; error bars are 95% image-bootstrap intervals. (c) Three-run score-only Top-1, with mean ± sample SD and evaluation-time controls.

Layerwise results. Both models exhibit the raw > scalar > vector loss ordering in every layer, including after zero queries are excluded (Table 4). No vector-recovery collision is detected in any evaluated row. In the final S layer, raw and scalar losses are 0.881 and 0.515, with 95% image-bootstrap intervals $[0.880, 0.882]$ and $[0.513, 0.517]$, respectively. For L, the corresponding losses are 0.870 and 0.437, with intervals $[0.869, 0.870]$ and $[0.435, 0.438]$. Collision severity varies across layers and model sizes, while the recovery ordering is preserved. Figure 4(a,b) illustrates the S results.

Score-only class readout. We train identical MLP readouts on the final attention layer. Sixteen positions on a 4×4 sampling grid within the native 14×14 token grid define a 16×16 score submatrix per head. Concatenating the eight heads gives 2,048 input features for a $2048 \to 256 \to 100$ ReLU MLP. Only the readout is trained. We use AdamW with learning rate 10^{-3}, weight decay 10^{-4}, batch size 256, and at most 50 epochs. Features are standardized using the probe training split, and model selection uses development loss.

Vector recovery improves mean Top-1 over raw scores by 1.83 percentage points for S and 1.13 points for L (Table 5). The L vector results vary more across runs, and the improvement is not uniform across run indices. Scalar recovery reduces collision loss but lowers mean readout accuracy in both models, demonstrating that improved local separation need not yield better accuracy for this readout. Evaluation-time constant-feature controls give 1.00% accuracy; sample-shuffle controls average 1.18% for S and 0.96% for L, both near the 1% chance level. Each model size uses one trained backbone, and the repeated runs jointly vary recovery and probe randomness. These experiments examine local score separation and class readout on the retained Q/K interface, rather than full-model classification performance.

Table 4: Layerwise normalized distinct-key loss on the two frozen models. Scalar results average three fixed recovery seeds. Vector loss is zero in every evaluated row for both query sets and is omitted from the columns.

		All sampled queries		Nonzero queries	
Model	Layer	Raw	Scalar	Raw	Scalar
SDT-V3-S	block3.0	.682	.298	.681	.299
	block3.1	.676	.326	.675	.325
	block3.2	.646	.290	.615	.257
	block3.3	.677	.335	.612	.290
	block3.4	.692	.300	.690	.297
	block3.5	.756	.488	.755	.488
	block4.0	.901	.537	.901	.536
	block4.1	.881	.515	.880	.514
SDT-V3-L	block3.0	.754	.287	.754	.287
	block3.1	.719	.377	.681	.321
	block3.2	.889	.524	.873	.513
	block3.3	.892	.525	.892	.524
	block3.4	.905	.559	.905	.557
	block3.5	.899	.537	.899	.537
	block4.0	.896	.555	.894	.548
	block4.1	.870	.437	.870	.436

Table 5: Score-only test Top-1 accuracy (%) over three recovery/probe runs per frozen model. SD denotes the sample standard deviation of these runs, not variation across backbone-training seeds.

Model	Recovery	Run 0	Run 1	Run 2	Mean $\pm$ SD
SDT-V3-S	Raw	19.35	19.60	20.75	19.90 ± 0.75
	Scalar	19.30	18.90	17.20	18.47 ± 1.12
	Vector	21.80	21.85	21.55	21.73 ± 0.16
SDT-V3-L	Raw	18.65	17.65	19.10	18.47 ± 0.74
	Scalar	16.10	14.15	13.40	14.55 ± 1.39
	Vector	21.15	16.25	21.40	19.60 ± 2.90