

Attention with a Calibrated Off Switch: Causal FIR Key Smoothing, the Score-Shift Pathology, and Selective Silence in Rectified Attention (pn_fir_tau)

Maxim Zhukov

mcfax.m@gmail.com

Independent researcher <https://orcid.org/0009-0005-9238-3778>

Research Article

Keywords: attention normalization, sub-simplex attention, selective silence, attention sinks, causal FIR filtering, induction heads, Triton kernels, language modeling

Posted Date: October 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11435949/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Attention with a Calibrated Off Switch: Causal FIR Key Smoothing, the Score-Shift Pathology, and Selective Silence in Rectified Attention (`pn_fir_tau`)

Maxim Zhukov*

Independent Research

Reproducibility Package: `src/`, `experiments/exp1..6`,
`exp4b`, `exp4c`, `exp4e`, `exp4f`, `exp4g`, `exp4h`,
`results/`, `run_all_and_build_figures.py`

October 2026

Abstract

Softmax attention cannot say “nothing found”: a simplex-constrained head must distribute 1.000 units of context weight even when no key matches the query, and pre-trained models pay with attention sinks, massive outliers that degrade low-bit quantization [13, 2, 4], and RAG-style hallucinations on unanswerable queries. The off-switch mechanism itself is established prior art: ReLA replaces Softmax with ReLU and its heads “attend to nothing” [1]; Softpick scales rectified normalization to 340M–1.8B with a 0% sink rate [2]. The open question is the interaction layer audited here. A parameter-free 7-tap causal FIR low-pass filter ($\sigma = 2.0$) applied strictly to the key stream K lets one layer bind a queried key onto the adjacent value slot, lifting non-bijective single-layer associative recall from $44.5\% \pm 0.9\%$ to $100.0\% \pm 0.0\%$ and removing exact-decoy vulnerability on the legacy benchmark ($\Delta = 0.000$ at decoy noise $\epsilon \in \{0.05, 0.2, 0.5, 1.0\}$). The same smoothing shifts scores downward and starves rectification: fixed- $\tau=0$ power-norm falls to $64.4\% \pm 6.8\%$ with $91.3\% \pm 8.6\%$ negative scores and $44.5\% \pm 23.0\%$ dead rows; Softmax masks this shift (translation-invariant), rectification does not (Proposition 1). One learnable threshold τ per layer (`pn_fir_tau`) restores 100.0% recall and keeps unanswerable rows silent under pure cross-entropy ($\sum_j W_{t,j} = 0.002 \pm 0.002$ vs. 0.100 ± 0.005 for virtual sinks) reaches matched-precision WikiText-2 parity on 2,400 steps ($\leq 0.3\%$ across five variants, single seed). We release fused Triton kernels (no `exp()`, $O(N)$ attention VRAM) and a full measurement record, including an instrumented retraction of our initial end-to-end latency claim.

1 Introduction: Practical Motivation and Related Work

Why modify the core attention mechanism of a Transformer? In practical sequence modeling, standard scaled dot-product Softmax attention [7] forces practitioners to pay a steep architectural and computational tax for two basic operations: **(A) reading the value token that follows a matched key in context** (in-context learning and associative lookup), and **(B) doing nothing**

*ORCID: [0009-0005-9238-3778](https://orcid.org/0009-0005-9238-3778)

when a specialized attention head finds no relevant match in the context (residual-stream preservation in RAG, tool use, and memory lookup).

1.1 Practical Problem 1: Compressing Two-Layer Induction Heads into a Single Layer (`sm_fir` / `pn_fir_tau`)

Associative retrieval—locating a context key k^* that matches a query and routing the value v^* stored at an adjacent position $j^* + 1$ (`[... key A, value B ... query A] -> B`)—is the foundational mechanism of in-context learning in Large Language Models (LLMs), enabling models to copy variable names, track entity attributes, and extract facts from prompts [6]. In a standard Transformer, however, **a single token-level attention layer is structurally incapable of content-addressable key-value induction**: dot-product similarity $S_{t,j} = \langle Q_t, K_j \rangle / \sqrt{d_k}$ matches the query Q_t (encoding k^*) against the key vector K_{j^*} at the key slot j^* , whereas the target value token v^* resides at the adjacent slot $j^* + 1$, whose unfiltered key vector $K_{j^*+1} = W_K E(v^*)$ carries zero information about k^* whenever keys and values are independently paired. As proved by Olsson et al. [6], a standard Transformer must dedicate **at least two sequential layers** to solve this task: Layer 1 (*previous-token head*) shifts token j^* into the residual stream of position $j^* + 1$, and only in Layer 2 (*induction head*) can attention match Q_t against the composite representation at $j^* + 1$. In compact or edge Transformers (4–12 layers), spending early layers and residual-stream bandwidth just to shift adjacent tokens is an expensive waste of depth.

Applying a parameter-free causal Finite Impulse Response (FIR) low-pass filter strictly to the key stream K along the sequence dimension (the convolutional ancestors of this move are RWKV’s learnable 2-tap token shift [5] on the R/K/V streams and the searched depthwise convolutions of *Primer* [8]; ours is a fixed zero-parameter $K_f = 7$ half-Gaussian low-pass strictly on K) blends the representation of k^* at position j^* forward into position $j^* + 1$:

$$\tilde{K}_{j^*+1} = h_0 K_{j^*+1} + h_1 K_{j^*} + \sum_{m=2}^{K_f-1} h_m K_{j^*+1-m}. \quad (1)$$

By Eq. (1), $\tilde{K}_{j^*+1}$ now contains a weighted component $h_1 K(k^*)$ while both the query Q_t and the value stream $V_{j^*+1} = W_V E(v^*)$ remain unfiltered, so **a single attention head querying for k^* can attend directly to the value slot $j^* + 1$ and read out v^*** —lifting single-layer non-bijective recall from 44.5% (`sm`) to $100.0\% \pm 0.0\%$ (`sm_fir` and `pn_fir_tau`) at a latency cost of just +1.5%–2.0% in `sm_fir`.

1.2 Practical Problem 2: Preventing RAG Context Hallucinations and Eliminating Attention-Sink Outliers (`pn_fir_tau`)

In a multi-layer Transformer, the residual stream acts as a shared communication bus ($y_t = x_t + \sum_{j \leq t} W_{t,j} V_j$). Most specialized attention heads (e.g., heads looking up an entity attribute or an answer in a retrieved RAG passage) are only relevant on a small fraction of tokens; when a retrieved passage does not contain the queried fact (**unanswerable query**), the head *should stay completely silent* ($\sum_{j \leq t} W_{t,j} = 0$) so that the existing residual representation x_t is preserved uncorrupted. Because standard Softmax enforces $\sum_{j=0}^t W_{t,j} \equiv 1$ for every query t , it causes two major practical failures (recently modelled as Softmax’s two missing primitives, abstention and noise filtering [3]): (1)

RAG context hallucinations on unanswerable queries, where the head is forced to overwrite the residual state x_t with a weighted average of irrelevant context values V_j ; and (2) **Attention Sinks and quantization outliers** [13] [14], where pre-trained LLMs turn the initial BOS token ($t = 0$) into a “trash bin” by inflating its hidden activations into massive numerical outliers [4] ($W_V x_0 \approx 0$) that degrade INT8/INT4 post-training quantization [13] (a failure mode rectified normalizers remove at 340M–1.8B scale with a low-bit win [2]) and cause perplexity collapse in streaming KV caches whenever the initial sink tokens are evicted [14].

In the sparse and quiet attention literature, two families have been proposed:

1. **Simplex-projected sparse normalizers (Sparsemax [9], α -Entmax [10], [11]):** For $\alpha > 1$ and $p = \frac{1}{\alpha-1}$, α -Entmax computes $W_{t,j}^{\text{entmax}}(s) = \max(s_j - \tau^*(s), 0)^p$, where $\tau^*(s)$ is the unique Lagrange multiplier solving $\sum_{j=0}^t \max(s_j - \tau^*(s), 0)^p = 1$. Because $\tau^*(s + c\mathbf{1}) = \tau^*(s) + c$, Sparsemax ($\alpha = 2$) and Entmax ($\alpha = 1.5, 1.1$) are translation-invariant, but because $\sum_{j=0}^t W_{t,j} \equiv 1$ everywhere, they **can never produce an all-zero row** (zero selective silence), and concentrating weight onto the single least-negative mismatching token actually *increases* context hallucination rates (32.4%–34.4% vs. 27.2% for Softmax in Exp 5).
2. **Virtual-sink Softmax (Softmax_1 [12], Quiet Softmax [13], Attention Sinks [14]):** Appending a virtual sink logit τ yields $W_{t,j} = \frac{\exp(s_j)}{\sum_{m=0}^t \exp(s_m) + \exp(\tau)}$, breaking shift-invariance so $\sum_{j=0}^t W_{t,j} < 1$, but retaining strictly positive exponential tails $\exp(s_j) > 0$ across all context positions ($\sum_j W_j = 0.100 \pm 0.005$ leakage on absent keys in Exp 5).
3. **Rectified (Sub-Simplex) Normalizers — established prior art (ReLA [1], Softpick [2]):** ReLA [1] replaces Softmax with ReLU, observes sparsity without projection, reports heads that “attend to nothing,” and stabilises training via LayerNorm initialisation or gating; Softpick [2] trains a rectified, deliberately non-sum-to-one normalizer at 340M–1.8B with a 0% sink rate and better low-bit quantized accuracy than Softmax. `pn_fir_tau` differs from this line in one falsifiable respect: it couples a per-layer learnable threshold τ with a peak-driven silence gate, and the threshold exists to calibrate the score shift that causal key smoothing imposes — not merely as a stability device.

Given that map, the open problem addressed here is not whether attention can be silent — prior art answers that — but whether a rectified normalizer survives causal key smoothing. Our audit (Section 4) uncovers a striking paradox when training is dominated by positive retrieval examples: adding causal FIR on K to Softmax (`sm_fir`) lifts clean associative recall to $100.0\% \pm 0.0\%$, whereas adding the exact same causal FIR filter on K to uncalibrated rectified power-normalization (`pn_fir`, fixed $\tau = 0$) degrades recall to $64.4\% \pm 6.8\%$ (Exp 1) and $88.8\% \pm 9.2\%$ (Exp 2), while row-mean centering (`pn_fir_c`) makes row mass dependent on within-row dispersion rather than absolute score level.

Scope. This is a controlled mechanistic study of how causal FIR smoothing interacts with sub-simplex normalization; the 1.8M-parameter language model and the synthetic recall tasks are deliberately small sandboxes that isolate the mechanism, not evidence of LLM-scale behaviour. We frame the contribution as a calibrated diagnostic and tool, and treat scale-up as future work.

2 Mathematical Formulation and Score-Shift Pathology

Let $X \in \mathbb{R}^{B \times T \times D}$ be the input sequence with model dimension D , H heads, and head dimension $d_k = D/H$. Linear projections $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ have shape $\mathbb{R}^{B \times H \times T \times d_k}$.

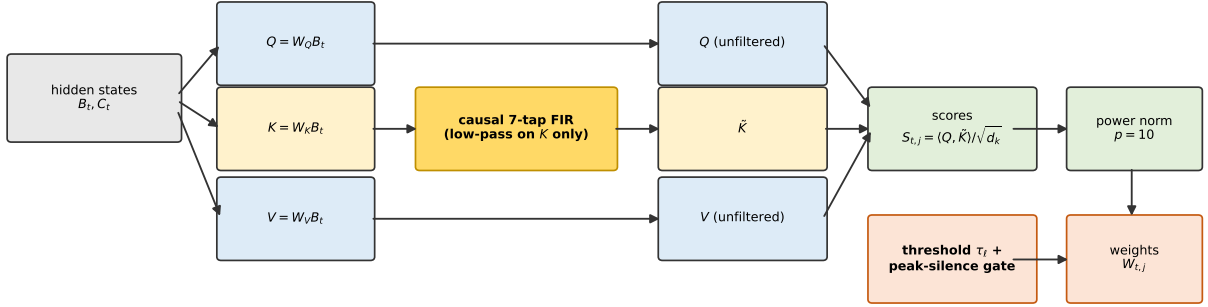
2.1 Causal FIR Low-Pass Filtering Strictly on the Key Stream K

Non-causal FFT filtering (`sm_fft`, `pn_fft`, `qe_pn_fft`) multiplies the full-sequence spectrum by a Gaussian mask, which leaks future tokens into past keys (`past_key_leak` = 0.785). To guarantee strict autoregressive causality (`past_key_leak` = 0.0), we apply a causal depthwise 1D FIR filter of length $K_f = 7$ exclusively to the key stream K , with half-Gaussian weights using $\sigma = \max((K_f - 1)/3.0, 0.5) = (7 - 1)/3.0 = 2.0$ (`causal_gaussian_taps` in `src/pnfirtau_attention.py`; Fig. 2):

$$h_m = \frac{\exp(-\frac{1}{2}(m/\sigma)^2)}{\sum_{r=0}^{K_f-1} \exp(-\frac{1}{2}(r/\sigma)^2)}, \quad m \in \{0, \dots, 6\}, \quad (2)$$

$$h = [0.3329, 0.2938, 0.2019, 0.1081, 0.0451, 0.0146, 0.0037].$$

With causal left-padding of $K_f - 1 = 6$ zeros using the taps in Eq. (2), $\tilde{K}_t = \sum_{m=0}^{\min(t,6)} h_m K_{t-m}$ depends exclusively on indices $\leq t$, while both Q and V remain unfiltered (Fig. 1), yielding scores $S_{t,j} = \langle Q_t, \tilde{K}_j \rangle / \sqrt{d_k}$ for $j \leq t$.



The filter acts on the key stream only: Q and V are left untouched, so the value slot next to a matched key is readable in one layer.

Figure 1: Schematic of the rectified attention head architecture. The causal FIR low-pass filter acts on the key stream only; the query and value streams are left untouched, so a matched key can be read at the adjacent value slot within a single layer. Source: `paper_en.tex` and `src/pnfirtau_attention.py`.

2.2 Simplex vs. Sub-Simplex Normalizers, Shift-Ray Invariance, and Score-Shift Pathology

Proposition 1 (Mass Constancy Along Uniform Shift Rays). *Let $\Phi : \mathbb{R}^n \rightarrow [0, 1]^n$ be any attention row normalizer with total row mass $M(s) = \sum_{j=1}^n \Phi_j(s) \in [0, 1]$.*

1. (**Simplex normalizers**): *If Φ maps onto the probability simplex ($M(s) \equiv 1$ for all $s \in \mathbb{R}^n$, as in Softmax, Sparsemax, and α -Entmax), then $M(s)$ is identically 1 everywhere and selective silence ($M(s) \rightarrow 0$) is impossible on any score vector $s \in \mathbb{R}^n$.*
2. (**Shift-invariant normalizers along uniform shift rays**): *If Φ is translation-invariant ($\Phi(s + c\mathbf{1}) = \Phi(s)$ for all $s \in \mathbb{R}^n, c \in \mathbb{R}$), then its total row mass is constant along every*

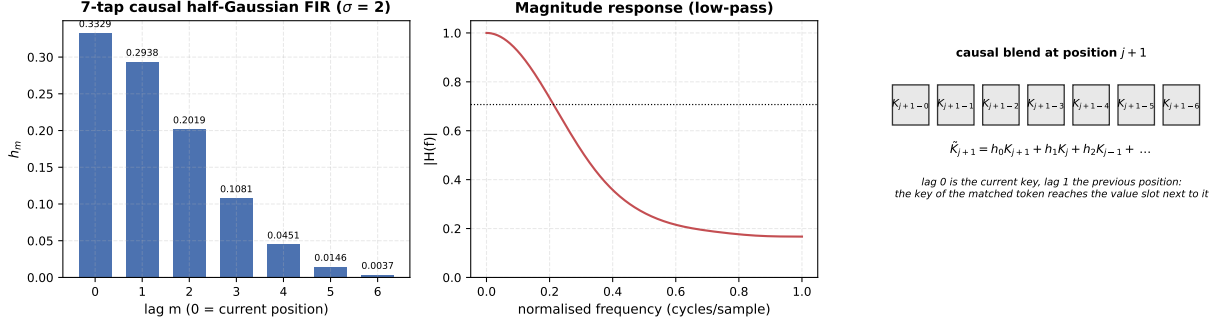


Figure 2: Diagram of the 7-tap causal FIR low-pass filter. Left: the tap coefficients, computed by the production function `causal_gaussian_taps` and normalised to sum to one. Centre: the magnitude response of the same taps. Right: how the causal blend mixes the current key with the preceding positions. The computed taps coincide with the coefficients printed in the paper. Source: `src/pnfirtau_attention.py` and `paper_en.tex`.

uniform shift ray $\{s - c\mathbf{1} : c \in \mathbb{R}\}$, i.e., $M(s - c\mathbf{1}) = M(s)$ for all $c \in \mathbb{R}$. Consequently, Φ cannot transition from active retrieval ($M(s) \approx 1$) to selective silence ($M(s - c\mathbf{1}) \rightarrow 0$ as $c \rightarrow +\infty$) when all scores in a row are shifted downward by a common offset c .

Why distinguish uniform shift rays from within-row dispersion (`pn_fir_c`)? Proposition 1(2) characterizes behavior along *uniform shift rays* $s \mapsto s - c\mathbf{1}$ (where within-row differences $s_j - s_m$ are preserved). A translation-invariant sub-simplex normalizer is not required to have constant mass across *all* $s \in \mathbb{R}^n$: in row-mean-centered power-normalization (`pn_fir_c`, $S'_{t,j} = S_{t,j} - \mu_t$ with peak gate $g_t = \frac{\max_j (S_{t,j} - \mu_t)_+}{\max_j (S_{t,j} - \mu_t)_+ + \gamma}$), $M(S_t) = g_t$ is invariant to uniform shifts $S_t \mapsto S_t - c\mathbf{1}$ because $\max_j S_{t,j} - \mu_t$ is shift-invariant, yet $M(S_t)$ depends on the **within-row score dispersion** $\max_j S_{t,j} - \mu_t$ (and thus on query norm $\|Q_t\|_2$). This explains all three empirical regimes of `pn_fir_c`: (a) without a peak gate ($\gamma = \text{None}$, Exp 2 Table 2), subtracting μ_t guarantees $\max_j (S_{t,j} - \mu_t) > 0$ on every non-constant row, forcing $\sum_j W_{t,j} \equiv 1.000$; (b) on uncalibrated noise rows in positive-only training ($\gamma = 0.5$, Exp 1 Table 1 and Exp 3 Table 5), random dot-product variance keeps $\max_j S_{t,j} - \mu_t \gg \gamma$, yielding $\sum W_{\text{noise}} = 0.938 \pm 0.007$ (gap $\leq +0.021$); and (c) when trained end-to-end with 50% unanswerable queries under Cross-Entropy loss ($\gamma = 0.5$, Exp 5 Table 6), `pn_fir_c` shrinks $\|Q_t\|_2$ on unanswerable queries to achieve partial silence ($\sum W_{\text{unans}} = 0.265 \pm 0.045$, gap $= +0.451 \pm 0.032$), but because $\max_j (S_{t,j} - \mu_t) > 0$ still holds on every non-constant row, it leaks $\approx 124\times$ more residual noise than `pn_fir_tau` (0.002 ± 0.002) while degrading answerable override accuracy ($88.5\% \pm 8.9\%$ vs. $99.0\% \pm 0.7\%$).

Because $\|h\|_2^2 = \sum_{m=0}^6 h_m^2 = 0.2518 < 1$ and each filtered key $\tilde{K}_j$ averages up to 7 adjacent tokens (most of which are negatively aligned filler tokens during positive-only training), the distribution of causal dot-product scores $S_{t,j} = \langle Q_t, \tilde{K}_j \rangle / \sqrt{d_k}$ contracts in variance and shifts negatively (Fig. 3):

- **Softmax (`sm_fir`), Sparsemax, and Entmax are translation-invariant:** Since $\text{softmax}(S_t - c) = \text{softmax}(S_t)$, Softmax is unaffected by negative offsets. In Exp 1, 100.0% of query rows in trained `sm_fir` have strictly negative scores (`raw_frac_pos` = 0.0%, `raw_rows_all_neg` = 100.0%), yet `sm_fir` achieves $100.0\% \pm 0.0\%$ accuracy.
- **Rectified power-norm (`pn_fir`) suffers dead-row starvation under positive-only training:** Uncalibrated power-norm rectifies scores at zero, $S_{t,j}^+ = \max(S_{t,j}, 0)$. When

$\max_{j \leq t} S_{t,j} \leq 0$, the entire row is clipped to zero, yielding $\hat{w}_{t,j} = 0$ and exact zero gradients ($\partial \mathcal{L} / \partial S_{t,j} = 0$). In Exp 1, $91.3\% \pm 8.6\%$ of raw scores in `pn_fir` are negative and $44.5\% \pm 23.0\%$ of query rows are completely dead (`raw_rows_all_neg` = $44.5\% \pm 23.0\%$ across seeds).

- **Row-mean centering (`pn_fir_c`) trades absolute level for within-row dispersion:** Subtracting the causal row mean $\mu_t = \frac{1}{t+1} \sum_{m=0}^t S_{t,m}$ forces roughly half of every noise row (44.4% of entries in Exp 1) above zero, driving noise attention mass to $\sum_j W_j = 0.938 \pm 0.007$ (Exp 1) and 1.000 ± 0.000 (Exp 2).

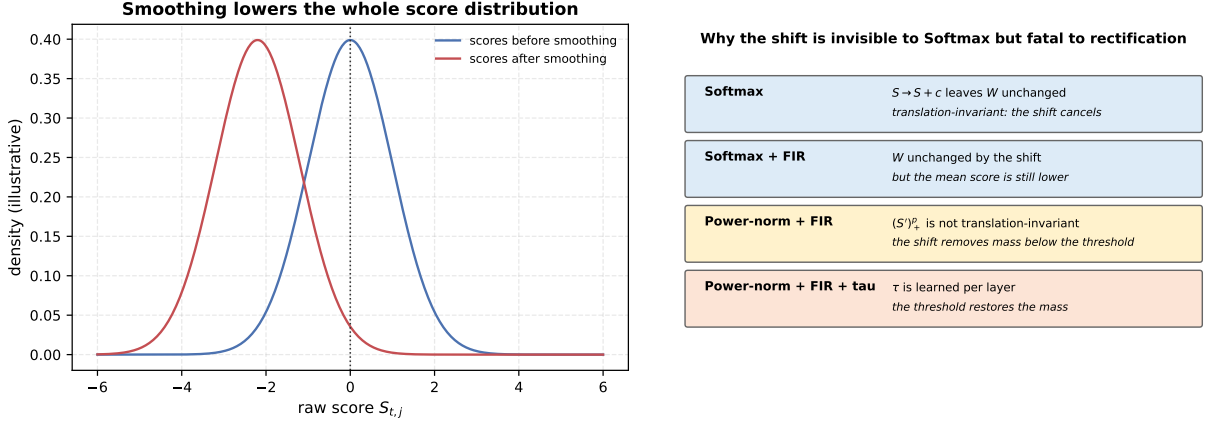


Figure 3: Diagram of the score-shift pathology. Left: smoothing lowers the whole score distribution, which a translation-invariant normaliser cancels but a rectified one does not. Right: the four regimes and why a negative shift is invisible to Softmax yet removes mass under a fixed threshold. The left panel is a schematic illustration of the mechanism; the measured shift is quantified in the text and in the width-ladder figure. Source: `paper_en.tex` (Proposition 1) and `src/pnfirtau_attention.py`.

2.3 The `pn_fir_tau` Formulation and IO-Aware Fused Triton Kernels

To calibrate the global score distribution without restoring within-row shift-invariance, `pn_fir_tau` introduces a single learnable scalar threshold $\tau \in \mathbb{R}$ per attention layer ($\tau_0 = 0.0$):

$$S'_{t,j} = (S_{t,j} - \tau) \cdot \mathbb{I}(j \leq t), \quad S_{t,j}^+ = \max(S'_{t,j}, 0), \quad (3)$$

$$\hat{w}_{t,j} = \frac{(S_{t,j}^+)^p}{\sum_{m=0}^t (S_{t,m}^+)^p + \varepsilon}, \quad g_t = \frac{\max_{0 \leq m \leq t} S_{t,m}^+}{\max_{0 \leq m \leq t} S_{t,m}^+ + \gamma}, \quad W_{t,j} = g_t \cdot \hat{w}_{t,j}, \quad (4)$$

where $p = 10.0$, $\varepsilon = 10^{-12}$, and $\gamma = 0.5$ is the peak silence gate constant in Eqs. (3)–(4). The total row weight is $\sum_{j=0}^t W_{t,j} = g_t \in [0, 1)$.

In eager PyTorch (`PnFirTauMHA`), evaluating Eqs. (3)–(4) materializes four $[B, H, N, N]$ intermediate tensors in HBM. In `src/triton_pnfirtau.py`, we eliminate this bottleneck via three custom Triton [15] kernel suites: (1) `triton_causal_fir` (zero-copy strided $K_f = 7$ causal FIR on K); (2) `PnFirTauTritonMHA` (`triton_pn_fir_tau_scores`, a single row-fused kernel replacing the 6-op eager score-to-weight chain); and (3) `PnFirTauFlashMHA` (`flash_pn_fir_tau_attention`, an online-tiled $O(N)$ -VRAM FlashAttention-2 style kernel with uniform tile size $B_M = B_N = 64$). Crucially, once fused in SRAM registers, $(S'_{t,j})_+^{10}$ is evaluated in 3 register multiplications ($u^2, u^4, u^8 \cdot u^2$) with

zero transcendental $\exp()$ calls, accumulates across tiles without online max-rescaling, computes exact analytical gradients in a single KV-parallel backward pass using the identity $D_t \triangleq \sum_{j=0}^t dW_{t,j} \hat{w}_{t,j} = \langle dO_t, O_t \rangle / g_t$ with deterministic per-block FP64 reduction for $\partial \mathcal{L} / \partial \tau$, and includes early-exit tile branches when $\max u_{t,j} \leq 0$ across a 64×64 tile (evaluated in Table 10).

3 Experimental Results and Algorithmic Audit

The experiments below are presented in dependency order, not numerical order: $1 \rightarrow 2 \rightarrow 6 \rightarrow 4f/4g \rightarrow 3 \rightarrow 5 \rightarrow 4$.

3.1 Experiment 1: Legacy 13-Stack Benchmark and Shortcut Audit

Table 1 reports the 3-seed evaluation (1234, 2025, 3141) of all 13 canonical stacks and 4 un-gated ($\gamma = \text{None}$) ablations on the legacy synthetic task ($V = 24, D = 32, T = 15$, 2,000 epochs, where $\text{KEY} \in \{1..5\}$ is at $t = 0$, $\text{VAL} = \text{KEY} + 5 \in \{6..10\}$ is at $t = 1$, filler $\in \{1..23\}$ is at 2..13, and $\text{QUERY} = \text{KEY}$ is at $t = 14$).

Table 1: Experiment 1: Full 13-stack legacy benchmark and 4 un-gated ablations (3 seeds: 1234, 2025, 3141, mean $\pm$ std). $W_{t,0}$ and $W_{t,1}$ denote query attention weight on the key slot ($t = 0$) vs. value slot ($t = 1$). Filter labels omit the default peak gate ($\gamma = 0.5$); the last four rows use $\gamma = \text{None}$. Source: `results/exp1_legacy_13stacks.json`.

Stack	Filter	Clean (%)	KR (%)	d@0.05	d@0.5	d@1.0	nd1/8 (%)	$\sum W_m$	$\sum W_n$	Gap	$W_{t,0}$	$W_{t,1}$
sm	None	71.5 $\pm$ 1.4	45.1	91.7	89.1	86.8	91.4/100.0	1.000	1.000	+0.000	0.130	0.136
sm_fft	FFT (Non-Causal)	99.0 $\pm$ 0.1	49.8	99.1	99.2	99.2	99.1/99.8	1.000	1.000	+0.000	0.219	0.068
sm_fir	FIR-7	100.0 $\pm$ 0.0	21.9	100.0	100.0	100.0	100.0/100.0	1.000	1.000	+0.000	0.984	0.013
pn	None	50.6 $\pm$ 3.9	43.2	57.9	55.6	53.0	57.2/74.6	0.973	0.955	+0.001	0.074	0.227
pn_c	Center	58.1 $\pm$ 8.6	38.2	72.3	68.1	63.3	71.6/77.3	0.993	0.988	+0.000	0.119	0.119
pn_tau	τ	55.9 $\pm$ 9.7	42.7	66.7	64.9	62.0	66.1/80.1	0.848	0.811	+0.002	0.085	0.154
pn_fft	FFT (Non-Causal)	98.9 $\pm$ 0.3	48.7	99.0	99.0	99.0	99.0/99.4	0.995	0.995	+0.000	0.222	0.075
pn_fft_c	FFT+C	81.6 $\pm$ 9.3	20.9	80.8	80.9	80.6	80.8/83.2	0.952	0.952	+0.002	0.256	0.028
pn_fir	FIR	64.4 $\pm$ 6.8	20.5	65.5	64.9	64.7	65.3/69.1	0.512	0.000	+0.375	0.389	0.001
pn_fir_c	FIR+C	100.0 $\pm$ 0.0	22.3	100.0	100.0	100.0	100.0/100.0	0.944	0.938	+0.017	0.904	0.037
pn_fir_tau	FIR+τ	100.0 $\pm$ 0.0	20.6	100.0	100.0	100.0	100.0/100.0	0.570	0.375	+0.191	0.554	0.012
qe_pn	QE+PN	71.2 $\pm$ 1.1	45.2	91.5	87.8	83.8	90.8/100.0	0.941	0.915	+0.000	0.136	0.140
qe_pn_fft	QE+PN+FFT	99.6 $\pm$ 0.0	41.4	99.7	99.7	99.7	99.7/99.0	0.913	0.907	+0.003	0.243	0.074
pn_ungated	None ($\gamma=\text{None}$)	49.7 $\pm$ 4.1	35.5	62.9	58.1	53.7	61.8/78.1	0.999	0.992	+0.001	0.136	0.152
pn_fir_ungated	FIR ($\gamma=\text{None}$)	71.7 $\pm$ 19.5	21.3	71.8	71.7	71.8	71.8/72.5	0.564	0.000	+0.517	0.414	0.092
pn_fir_c_ungated	FIR+C ($\gamma=\text{None}$)	100.0 $\pm$ 0.0	18.7	100.0	100.0	100.0	100.0/100.0	1.000	1.000	+0.000	0.998	0.002
pn_fir_tau_ungated	FIR+ τ ($\gamma=\text{None}$)	100.0 $\pm$ 0.0	18.7	100.0	100.0	99.9	100.0/100.0	1.000	0.717	+0.282	0.999	0.001

Audit of Exp 1 Shortcuts. Inspection of $W_{t,0}$ vs. $W_{t,1}$ in Table 1 exposes two shortcuts in the legacy task: (1) because $\text{VAL} = \text{KEY} + 5$ is a fixed bijection and KEY is always at $t = 0$, left-padded causal FIR acts as a boundary detector for $t = 0$ ($W_{t,0} = 0.984$ for `sm_fir`, 0.554 of 0.570 for `pn_fir_tau`), predicting VAL directly from $V_0 = W_V E(\text{KEY})$ without reading $t = 1$; and (2) because clean filler at 2..13 draws from $\{1..23\}$ (which includes competing keys $\{1..5\}$), injecting $n_d = 8$ copies of $E(\text{KEY})$ overwrites competing keys with copies of the true key, explaining why `sm` rises from 71.5% (clean) to 100.0% ($n_d = 8$).

3.2 Experiment 2: True Content-Addressable Non-Bijective Recall & Distractor Decoys

To eliminate both shortcuts, Experiment 2 places $M = 3$ randomly paired (k_m, v_m) pairs ($k_m \in \{1..8\}$, independently permuted $v_m \in \{9..16\}$) at random non-overlapping interior positions $j_m \in [2, 38]$ in $T = 48$, with filler strictly from $\{17..23\}$. True distractor decoys overwrite $n_d \in \{1, 2, 4, 8\}$ isolated filler slots with $(1 - \epsilon)E(k^*) + \epsilon E(\text{filler})$. We also evaluate `pn_fir_tau_reg`, which adds a post-warmup (epoch ≥ 1000) silence penalty $\lambda_{\text{sil}} = 0.05$ on auxiliary absent-key queries.

Table 2: Experiment 2: True Content-Addressable Non-Bijective Associative Recall ($M = 3$ pairs in $T = 48$, 3 seeds: 1234, 2025, 3141, mean $\pm$ std). Here W_{t,j^*} and W_{t,j^*+1} measure attention on the interior target key slot vs. target value slot. Source: `results/exp2_content_addressable_decoy.json`.

Model	Clean (%)	$\epsilon=0.05$	$\epsilon=0.2$	$\epsilon=0.5$	$\epsilon=1.0$	nd1/2/4/8	W_{t,j^*}	W_{t,j^*+1}	$\sum W_m$	$\sum W_n$	Gap
sm	44.5 $\pm$ 0.9	47.6	44.5	43.8	46.5	45.5/45.7/44.6/46.8	0.035	0.234	1.000	1.000	+0.000 $\pm$ 0.000
pn	35.8 $\pm$ 1.0	35.9	35.4	34.4	35.1	35.1/36.6/36.1/35.8	0.015	0.237	1.000	1.000	+0.000 $\pm$ 0.000
sm_fir	100.0 $\pm$ 0.0	95.9	95.9	95.2	92.0	95.1/89.9/77.7/56.2	0.132	0.792	1.000	1.000	+0.000 $\pm$ 0.000
pn_fir	88.8 $\pm$ 9.2	85.9	85.5	84.7	82.1	84.1/80.0/71.0/61.2	0.032	0.830	1.000	0.261	+0.739 $\pm$ 0.117
pn_fir_c	96.5 $\pm$ 4.9	93.7	93.4	92.8	90.1	92.5/88.9/78.2/63.2	0.044	0.866	1.000	1.000	+0.000 $\pm$ 0.000
pn_fir_tau	100.0 $\pm$ 0.0	95.5	95.8	95.5	93.1	95.6/90.5/74.7/56.9	0.063	0.729	0.826	0.218	+0.608 $\pm$ 0.135
pn_fir_tau_reg	100.0 $\pm$ 0.0	93.8	93.1	91.8	87.4	93.1/82.6/60.6/38.8	0.034	0.647	0.684	0.000	+0.684 $\pm$ 0.030

Table 2 and Figures 4–5 confirm that when the bijection and boundary shortcuts are removed: (1) unfiltered `sm` (44.5%) and `pn` (35.8%) fail; (2) causal FIR on K physically binds $h_1 K(k^*)$ onto the value slot $j^* + 1$, shifting attention directly to $j^* + 1$ ($W_{t,j^*+1} = 0.792$ for `sm_fir`, 0.729 for `pn_fir_tau`, 0.647 for `pn_fir_tau_reg`); and (3) `pn_fir_tau_reg` learns $\tau = +0.115 \pm 0.021$, driving both noise and absent-key attention mass to 0.000 ± 0.000 across all seeds (gap = $+0.684 \pm 0.030$).

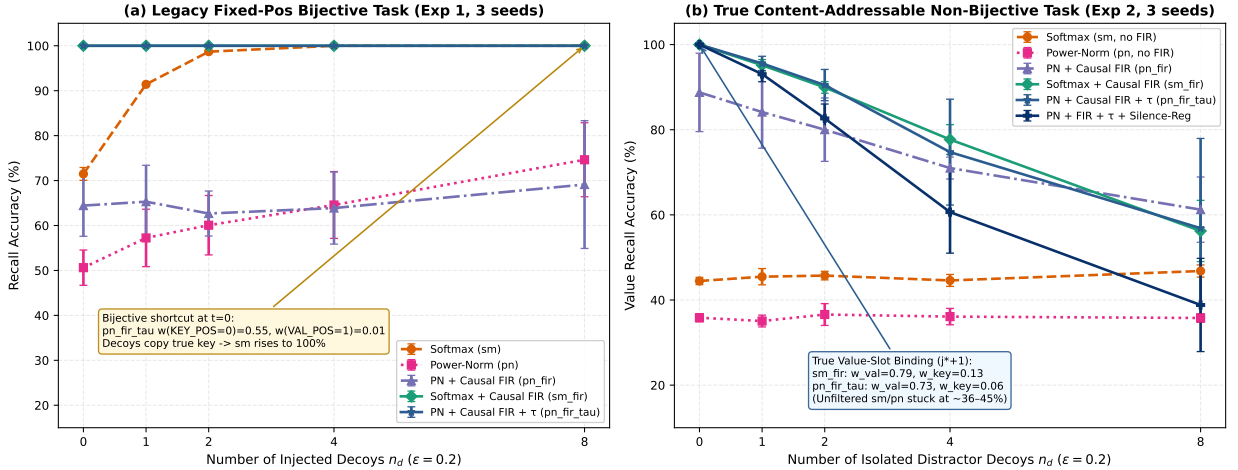


Figure 4: Multi-decoy scaling ($n_d \in \{0, 1, 2, 4, 8\}$) and attention slot routing on the legacy fixed-position task (Exp 1, left) vs. the true content-addressable non-bijective task (Exp 2, right). Source: `results/exp1_legacy_13stacks.json` and `results/exp2_content_addressable_decoy.json`.

3.3 Experiment 6: Token-Shift Ablation (How Much Memory Does Robustness Need)?

Exp 2’s flooding result raises a precise follow-up: is the 7-tap low-pass doing work a 2-tap RWKV-style token shift cannot? `experiments/exp6_token_shift_ablation.py` reruns both decoy proto-

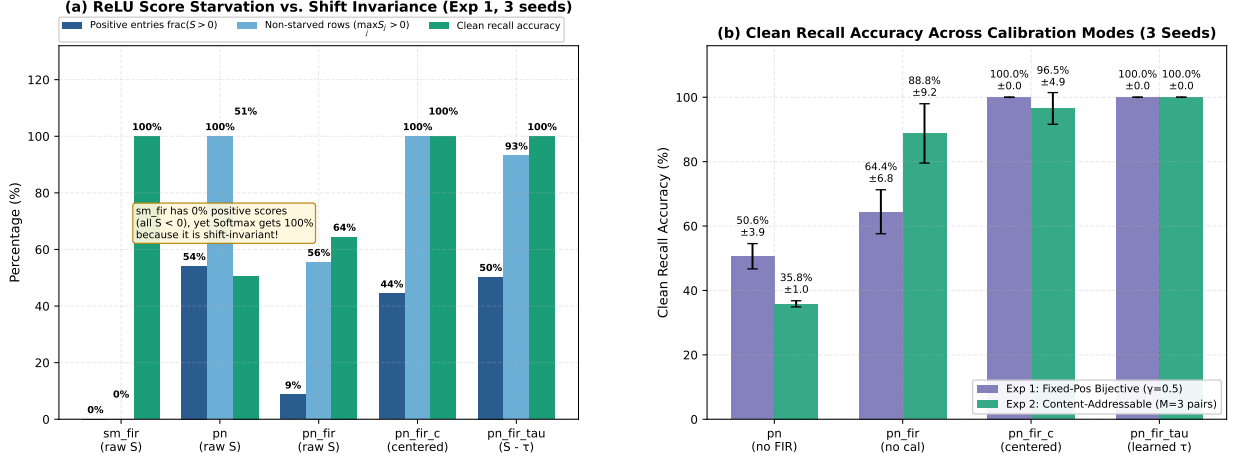


Figure 5: Pre-activation score-shift pathology in Exp 1 (left) and 3-seed clean recall accuracy across calibration modes in Exp 1 and Exp 2 (right). Source: `results/exp1_legacy_13stacks.json` and `results/exp2_content_addressable_decoy.json`.

cols with the filter on K swapped for a fixed 2-tap shift $[0.5, 0.5]$ and a learnable per-channel μ shift, at the exact FIR position (placement fixed; pre-projection placement untested). Controls are rerun in-script (`sm/sm_fir` reproduce Exp 1 bit-for-bit); all shift arms pass the causal probe.

Table 3: Token-shift ablation (3 seeds; `exp6`). Binding needs lag-1 memory; flooding favours shorter memory; rectified short-memory arms are seed-unstable.

Arm	Exp 1 clean	Exp 1 d@1.0	Exp 1 nds	Exp 2 clean	Exp 2 eps _{0.2}	Exp 2 nds	Exp 2 W_{val}
sm	71.5 $\pm$ 1.4	86.8 $\pm$ 1.3	100.0 $\pm$ 0.0	44.5 $\pm$ 0.9	44.5 $\pm$ 1.8	46.8 $\pm$ 1.4	0.234 $\pm$ 0.014
sm_fir (7-tap)	100.0 $\pm$ 0.0	100.0 $\pm$ 0.0	100.0 $\pm$ 0.0	100.0 $\pm$ 0.0	95.9 $\pm$ 0.7	56.2 $\pm$ 7.2	0.792 $\pm$ 0.011
sm_shiftmu ($\mu=0.465$)	100.0 $\pm$ 0.0	96.9 $\pm$ 0.6	100.0 $\pm$ 0.0	100.0 $\pm$ 0.0	100.0 $\pm$ 0.0	99.9 $\pm$ 0.2	0.970 $\pm$ 0.002
pn_fir_tau (7-tap)	100.0 $\pm$ 0.0	99.7 $\pm$ 0.4	100.0 $\pm$ 0.0	100.0 $\pm$ 0.0	95.7 $\pm$ 2.0	56.7 $\pm$ 21.1	0.719 $\pm$ 0.021
pn_shiftmu_tau	76.5 $\pm$ 10.9	77.1 $\pm$ 11.1	81.8 $\pm$ 9.3	92.2 $\pm$ 11.0	92.3 $\pm$ 11.0	92.7 $\pm$ 10.4	0.798 $\pm$ 0.094

Three findings, summarized in Table 3 and Fig. 6. **(1) Binding needs only lag-1 memory:** both shift variants reach 100.0% clean recall on both protocols; learned μ pins at 0.465 ± 0.002 on Exp 2 (channels $[0.451, 0.492]$) but wanders on the legacy task ($[0.366, 0.600]$), consistent with its boundary shortcut. **(2) Flooding favours shorter memory:** at $n_d=8$, `sm_shiftmu` holds $99.9\% \pm 0.2\%$ vs. $56.2\% \pm 7.2\%$ for 7-tap `sm_fir` ($W_{\text{val}} = 0.970$ vs. 0.792). **(3) For rectified attention the 7-tap filter is the stable partner of a single scalar τ :** short-memory rectified arms are seed-unstable (`pn_shiftmu_tau` clean $76.5\% \pm 10.9\%$ on Exp 1 with per-seed τ - μ equilibria); the 7-tap filter’s large consistent shift is reliably absorbed by one scalar.

3.4 Experiment 4f-A/4g: Width Envelope and the Tau Fixed Point

The pathology amplifies with width rather than attenuating (ladder $d \in \{16, 24, 32, 48, 64, 96, 128\}$, identical recipe, 3 seeds; `exp4f_scaleout.py` Part A and `exp4g_tau_scaling_law.py`; Fig. 7 and Fig. 8; Table 4 lists the ladder values). `sm_fir` holds 100.0% at every width (its raw shift drifts mildly, $-14 \rightarrow -27$); uncalibrated `pn_fir` raw magnitudes diverge to tens and hundreds with seed-divergent signs ($\mu_S = -20 \dots -144$, σ_S up to ~ 300), and `pn_fir_tau` converges fully only at $d \in \{32, 48, 64\}$ (collapsing to 52.6% at $d=128$; partial at 16/24, stochastic at 96/128). A linear drift-cancellation law $\tau^*(d) = a \cdot (-\mu_S) + b$ is rejected ($R^2 = 0.45$, slope ≈ 0); instead the calibrated fixed point is width-invariant ($\tau^* \in [-1.4, -0.7]$ wherever convergence happens) while only its basin

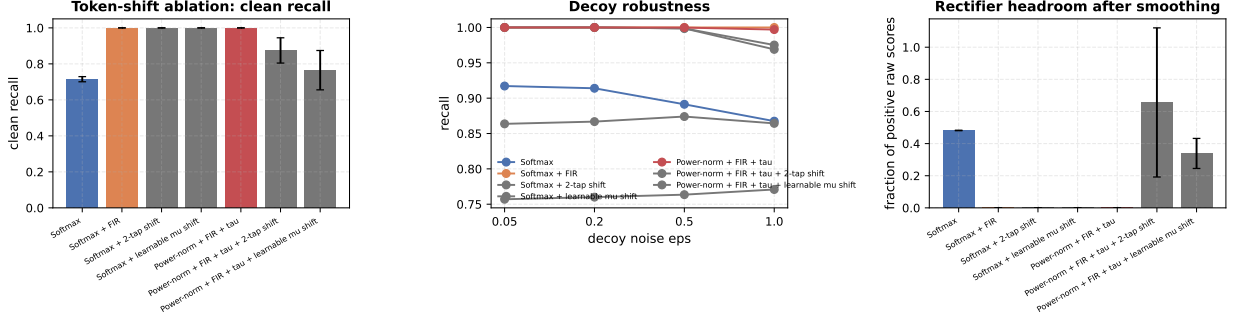


Figure 6: Token-shift ablation on the legacy protocol. Left: clean recall per variant. Middle: recall against the decoy-noise level. Right: the fraction of positive raw scores, i.e. the headroom the rectifier has left after smoothing. Bars and curves are means over $n = 3$ seeds; error bars are the standard deviation over the same seeds. Source: `results/exp6_token_shift_ablation.json` (`partA_legacy.summary`).

from $\tau_0 = 0$ collapses. Initializing the predicted $\tau_0 = -1.02$ rescues $d=128$ on all 3 seeds (100.0%, equilibria $-1.19/-1.16/-1.16$) — a dead-row basin problem at initialization, not a representational wall. Practical rule: start τ negative at width. (The H-gap diagnostic, $H_{\text{noise}} - H_{\text{match}}$, is ≈ 0 at every width including converged ones — uninformative under $p=10$ sharpening, a negative result on our own design.)

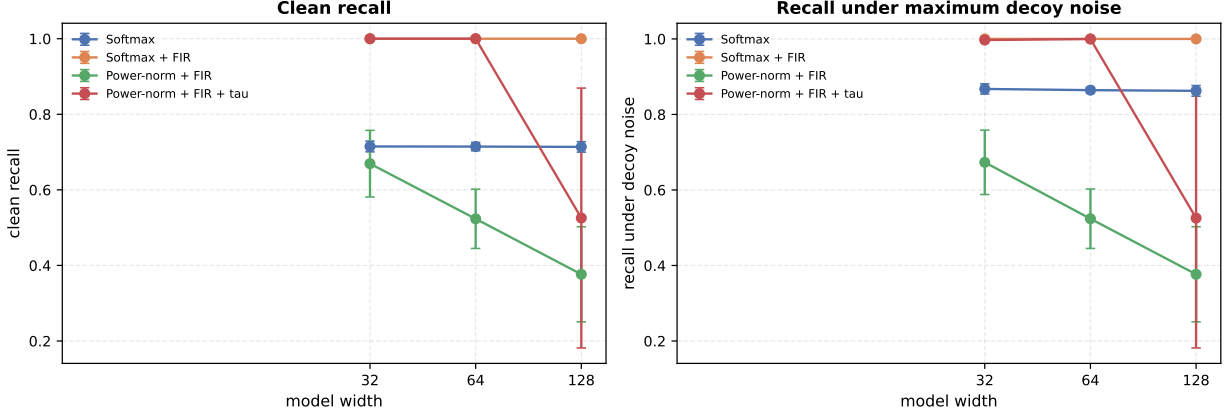


Figure 7: Width envelope of the four attention variants on the legacy task. Left: clean recall; right: recall under the strongest decoy noise. Points are means over $n = 3$ seeds; error bars are the standard deviation over the same seeds. Source: `results/exp4f_scaleout.json` (`partA.arms.<arm>.by_width.<width>`).

3.5 Experiment 3: 31-Configuration Sweep of $(\tau_0, \gamma, \text{silence_mode})$

Table 5 and Figure 9b summarize the 31-configuration sweep across threshold initialization τ_0 , gate constant γ , and gating mode (`peak` vs. `mass_relu`). Initializing $\tau_0 = +0.5$ causes immediate ReLU starvation (clean = 20.6%), while $\tau_0 = 0.0$ with `peak` gating ($\gamma \in [0.1, 0.5]$) achieves 100.0% recall and strong silence (gap = +0.572 at $\gamma = 0.10$ and +0.432 at $\gamma = 0.50$ on seed 1234). In contrast, `mass_relu` (gap $\leq +0.010$) and `pn_fir_c` on positive-only training (gap $\leq +0.021$, because uncalibrated noise rows have large within-row dispersion $\max_j S_{t,j} - \mu_t \gg \gamma$) fail to suppress noise across all γ .

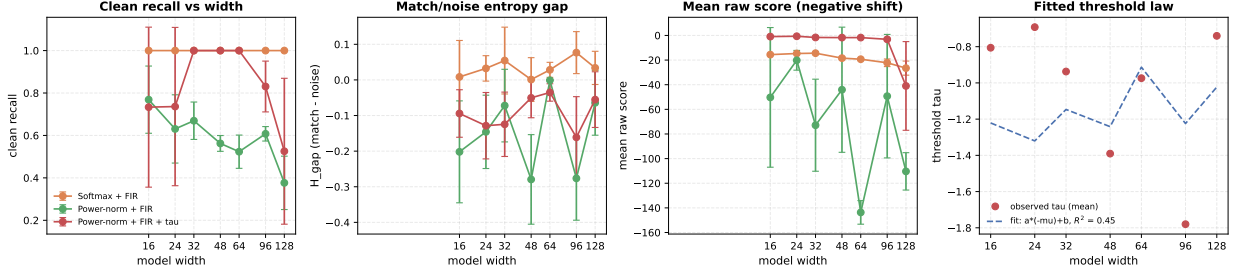


Figure 8: Behaviour of the three filtered variants across a seven-point width ladder, with the fitted threshold law. From left to right: clean recall, the match/noise entropy gap, the mean raw score, and the learned threshold against the fitted law $\tau = a \cdot (-\mu_S) + b$ ($R^2 = 0.45$, fitted on widths 32, 48, 64). Points are means over $n = 3$ seeds; error bars are the standard deviation over the same seeds. Source: `results/exp4g_tau_scaling_law.json` (ladder, fit).

Table 4: Experiment 4f-A: width envelope of the four attention variants on the legacy task (ladder $d \in \{16, 24, 32, 48, 64, 96, 128\}$, $n = 3$ seeds). μ_S is the mean raw score of the uncalibrated `pn_fir` arm, τ^* the learned threshold of `pn_fir_tau`, and the H-gap the match/noise entropy difference. The last two rows report the predicted negative initialisation $\tau_0 = -1.02$ and a $10\times$ smaller learning rate at $d = 128$. Source: `results/exp4f_scaleout.json` and `results/exp4g_tau_scaling_law.json`.

Width d	sm_fir clean	pn_fir clean	pn_fir raw μ_S	pn_fir_tau clean	pn_fir_tau τ^*	H-gap (tau arm)
16	100.0	76.9	-50.3	73.3	-0.81 ± 0.60	-0.09
24	100.0	63.1	-20.2	73.6	-0.69 ± 0.52	-0.13
32	100.0	66.9	-72.9	100.0	-0.94 ± 0.02	-0.13
48	100.0	56.2	-44.1	100.0	-1.39 ± 0.48	-0.05
64	100.0	52.3	-143.7	100.0	-0.97 ± 0.05	-0.04
96	100.0	60.8	-49.3	83.1	-1.78 ± 0.74	-0.16
128	100.0	37.7	-110.4	52.6	-0.74 ± 0.19	-0.06
128, $\tau_0 = -1.02$ (predicted)	—	—	—	100.0	-1.17 ± 0.02	-0.02
128, lr = 10^{-3}	100.0	—	—	91.0 ± 12.8	-0.60 ± 0.14	—

Table 5: Experiment 3: Representative configurations from the 31-point ($\tau_0, \gamma, \text{mode}$) sweep. Source: `results/exp3_sweep_tau_gate.json`.

Stack and Mode	τ_0	γ	Clean	$\sum W_m$	$\sum W_n$	Gap	τ	3-Seed Clean	3-Seed $\sum W_n$	3-Seed Gap
pn_fir_tau (peak)	0.0	None	100.0%	1.000	0.752	+0.228	-0.320	100.0%	0.717	+0.282 $\pm$ 0.045
pn_fir_tau (peak)	0.0	0.05	100.0%	0.882	0.608	+0.221	-0.453	100.0%	0.628	+0.197 $\pm$ 0.054
pn_fir_tau (peak)	0.0	0.10	100.0%	0.572	0.000	+0.572	-0.501	100.0%	0.336	+0.373 $\pm$ 0.180
pn_fir_tau (peak)	0.0	0.20	100.0%	0.752	0.501	+0.211	-0.696	100.0%	0.583	+0.129 $\pm$ 0.070
pn_fir_tau (peak)	0.0	0.50	100.0%	0.436	0.003	+0.432	-0.940	100.0%	0.375	+0.191 $\pm$ 0.171
pn_fir_tau (peak)	-0.5	0.50	100.0%	0.672	0.433	+0.193	-1.159	100.0%	0.484	+0.165 $\pm$ 0.029
pn_fir_tau (peak)	-1.0	0.50	100.0%	0.724	0.611	+0.108	-1.531	100.0%	0.593	+0.122 $\pm$ 0.019
pn_fir_tau (peak)	+0.5	0.50	20.9%	0.000	0.000	+0.000	+0.500	20.6%	0.000	+0.000 $\pm$ 0.000
pn_fir_c (peak)	0.0	0.50	100.0%	0.947	0.946	+0.021	N/A	100.0%	0.938	+0.017 $\pm$ 0.004
pn_fir_tau (mass_relu)	0.0	0.50	100.0%	0.847	0.851	-0.001	-0.481	100.0%	0.815	+0.010 $\pm$ 0.008
pn_fir_tau (mass_relu)	0.0	1.00	100.0%	0.793	0.784	+0.003	-0.613	100.0%	0.776	+0.008 $\pm$ 0.006

3.6 Experiment 5: Context Override vs. Residual Stream Preservation Against Sparsemax, Entmax, and Softmax Sinks Under Pure Cross-Entropy Loss

To test whether sub-simplex selective silence yields a concrete end-to-end task advantage over simplex normalizers (`sm_fir`, Sparsemax [9], and α -Entmax [10] [11]), row-mean-centered power-norm (`pn_fir_c`), and virtual-sink Softmax (`softmax1_fir`, `softmax_sink_fir`) **under 100% pure task Cross-Entropy loss with zero auxiliary attention regularization**, Experiment 5 (`exp5_abstention_and_entmax.py`) evaluates a topic-coherent document residual stream task ($T = 48$, $M = 3$ pairs $(k_m, v_m) \in \{0..7\}^2$, filler tokens carrying topic values from v_{sub} , and

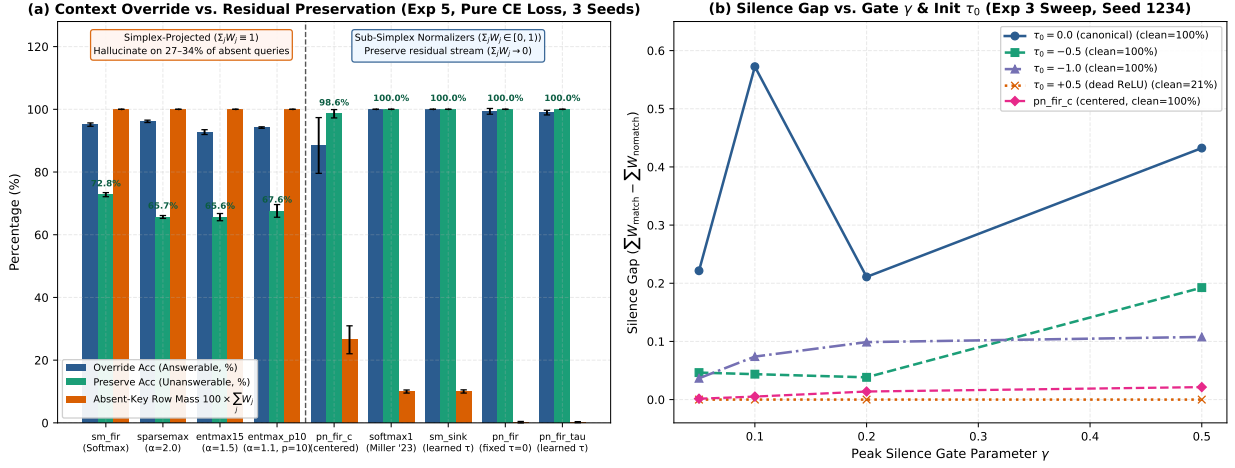


Figure 9: Context override vs. residual stream preservation across simplex and sub-simplex normalizers under pure Cross-Entropy loss (Exp 5, left), and silence gap vs. gate γ and initialization τ_0 (Exp 3 sweep, right). Source: `results/exp5_abstention_and_entmax.json` and `results/exp3_sweep_tau_gate.json`.

query residual update $y = V(v_{\text{default}}) + 2.5 \sum_{j < 47} W_{\text{query},j} V_j$ where $v_{\text{default}} \notin v_{\text{sub}}$). On 50% of queries (k^* present), the head must override v_{default} with v^* ; on 50% of queries (k^* absent), the head must stay silent ($\sum_j W_{\text{query},j} \rightarrow 0$) to preserve v_{default} .

Table 6: Experiment 5: Context Override vs. Residual Preservation under pure Cross-Entropy loss (3 seeds: 1234, 2025, 3141, mean $\pm$ std). All 9 models use the 7-tap causal FIR filter on K . Source: `results/exp5_abstention_and_entmax.json`.

Family	Model	Overall (%)	Overr. (%)	Pres. (%)	Hall. (%)	$\sum W_a$	$\sum W_u$	Gap	W_{t,j^*+1}
Simplex	sm_fir	84.0 ± 0.6	95.1 ± 0.5	72.8 ± 0.7	27.2 ± 0.7	1.000	1.000	+0.000	0.538
Simplex	sparsemax_fir ($\alpha = 2.0$)	80.9 ± 0.2	96.2 ± 0.4	65.7 ± 0.4	34.3 ± 0.4	1.000	1.000	+0.000	0.604
Simplex	entmax15_fir ($\alpha = 1.5$)	79.2 ± 0.8	92.7 ± 0.8	65.6 ± 1.1	34.4 ± 1.1	1.000	1.000	+0.000	0.593
Simplex	entmax_p10_fir ($\alpha = 1.1$)	80.9 ± 0.9	94.2 ± 0.2	67.6 ± 2.0	32.4 ± 2.0	1.000	1.000	+0.000	0.560
Centered	pn_fir_c	93.5 ± 5.1	88.5 ± 8.9	98.6 ± 1.3	1.4 ± 1.3	0.716	0.265	+0.451	0.577
Virtual Sink	softmax1_fir ($\tau = 0$)	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	0.0 ± 0.0	0.863	0.100	+0.764	0.776
Virtual Sink	softmax_sink_fir (τ)	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	0.0 ± 0.0	0.864	0.100	+0.764	0.775
Sub-Simplex	pn_fir ($\tau = 0, p = 10$)	99.7 ± 0.5	99.4 ± 0.9	100.0 ± 0.0	0.0 ± 0.0	0.747	0.002	+0.745	0.724
Sub-Simplex	pn_fir_tau ($\tau, p = 10$)	99.5 ± 0.4	99.0 ± 0.7	100.0 ± 0.0	0.0 ± 0.0	0.744	0.002	+0.742	0.716

Table 6 and Figure 9a establish four key findings: (1) all four simplex normalizers fail residual preservation (27.2%–34.4% hallucination rate), with `sparsemax_fir` and `entmax` performing even worse than `sm_fir` because sparse simplex projection concentrates weight onto a single mismatching topic token; (2) under pure Cross-Entropy loss with 50% unanswerable queries, `pn_fir_tau` places unanswerable raw score peaks below $\tau = +0.011 \pm 0.020$ ($\max_j S_{\text{unans}}^{\text{raw}} = -0.175 \pm 0.070 < \tau$, while $\max_j S_{\text{ans}}^{\text{raw}} = +1.548 \pm 0.281 \gg \tau$), achieving $100.0\% \pm 0.0\%$ residual preservation (0.0% hallucination rate) and $\sum W_{\text{unans}} = 0.002 \pm 0.002$ (note that because 50% of training examples are unanswerable, W_Q, W_K naturally center unanswerable scores below zero, so fixed $\tau = 0$ `pn_fir` performs equally well here at $99.7\% \pm 0.5\%$, whereas learnable τ is essential when training is dominated by positive retrieval as in Exp 1–2 or multi-layer LM in Exp 4); (3) `pn_fir_c` shrinks $\|Q_{\text{query}}\|_2$ on unanswerable rows to reach $\sum W_{\text{unans}} = 0.265 \pm 0.045$ (gap = $+0.451 \pm 0.032$), but still leaks $\approx 124\times$ more residual noise than `pn_fir_tau` while degrading answerable override accuracy ($88.5\% \pm 8.9\%$ vs. $99.0\% \pm 0.7\%$); and (4) comparing `pn_fir_tau` against virtual-sink Soft-

`max (softmax1_fir, softmax_sink_fir)` reveals a clean trade-off: virtual-sink Softmax achieves slightly higher discrete answerable override accuracy ($100.0\% \pm 0.0\%$ vs. $99.0\% \pm 0.7\%$) and higher active mass ($\sum W_{\text{ans}} = 0.863\text{--}0.864$ vs. 0.744), whereas `pn_fir_tau` achieves $50\times$ lower continuous residual-stream contamination on unanswerable queries ($\sum W_{\text{unans}} = 0.002 \pm 0.002$ vs. 0.100 ± 0.005).

3.7 Experiment 4: 6-Layer Causal Language Modeling on WikiText-2 and Fused Triton Benchmarks

We evaluate a 6-layer Pre-LN Transformer on word-level WikiText-2 ($V = 23,168, D = 64, H = 4, d_{\text{ff}} = 256, N = 256, B = 64$, tied embeddings: $1,482,752$ (tok_emb) + $16,384$ (pos_emb) + $6 \times 49,728$ ($298,368$) + 128 (ln_f) = $1,797,632$ base parameters; +6 scalar τ_{ℓ} parameters in `pn_fir_tau` for $1,797,638$ total). In IEEE-754 FP32 parity checks (`allow_tf32=False`, reference norms $\|y\|_{\infty} = 0.5441, \|dx\|_{\infty} = 2.1698, |d\tau| = 7.0289$), both `PnFirTauTritonMHA` (`out_max_diff` = 2.38×10^{-7} [rel 4.38×10^{-7}], `dx_max_diff` = 1.16×10^{-6} [rel 5.36×10^{-7}], `dtau_diff` = 3.91×10^{-5} [rel 5.56×10^{-6}]) and `PnFirTauFlashMHA` (`out_max_diff` = 2.38×10^{-7} , `dx_max_diff` = 1.32×10^{-6} [rel 6.08×10^{-7}], `dtau_diff` = 4.01×10^{-5} [rel 5.70×10^{-6}]) closely match eager PyTorch. Table 7, Table 9, Table 10, and Figure 18 present the full benchmark suite.

Table 7: Experiment 4: WikiText-2 6-layer Language Model benchmarks ($B = 64, N = 256$, RTX 3060). Suites 1–3 and suite 2s are defined in the text; suite 3 is precision-controlled across 3 seeds; suite 2s ($D=256$) is a single-seed pilot; step times are omitted for suite 3 and suite 2s (see notes below). Source: `results/exp4_wikitext2_lm.json`.

Suite	Variant	Precision	ms/step	Rel. Time	VRAM (GB)
suite 1: Speed/VRAM	<code>softmax</code>	BF16+TF32	97.70	1.000×	5.16
suite 1: Speed/VRAM	<code>sm_fir</code> (FIR)	BF16+TF32	99.21	1.016×	5.17
suite 1: Speed/VRAM	<code>pn_fir_tau</code> (Eager)	BF16+TF32	164.44	1.683×	6.11
suite 1: Speed/VRAM	<code>pn_fir_tau_triton</code>	BF16+TF32	75.74	0.775×	4.97
suite 1: Speed/VRAM	<code>pn_fir_tau_flash</code>	BF16+TF32	66.80	0.684×	4.52
suite 1: Speed/VRAM	<code>softmax</code>	FP32+TF32	140.92	1.000×	6.92
suite 1: Speed/VRAM	<code>sm_fir</code> (FIR)	FP32+TF32	143.76	1.020×	6.94
suite 1: Speed/VRAM	<code>pn_fir_tau</code> (Eager)	FP32+TF32	244.17	1.733×	8.82
suite 1: Speed/VRAM	<code>pn_fir_tau_triton</code>	FP32+TF32	120.43	0.855×	6.92
suite 1: Speed/VRAM	<code>pn_fir_tau_flash</code>	FP32+TF32	105.84	0.751×	6.10

Suite	Variant	Precision	Params	Train	Val	Test	Leak
suite 2	<code>softmax</code>	BF16+TF32	1,797,632	654.23	547.33	515.55	5.16
suite 2	<code>sm_fir</code>	BF16+TF32	1,797,632	652.97	547.29	515.18	5.17
suite 2	<code>pn_fir_tau</code> (Eager)	BF16+TF32	1,797,638	653.17	546.75	515.56	6.11
suite 2	<code>pn_fir_tau_triton</code>	BF16+TF32	1,797,638	652.00	545.78	515.40	4.97
suite 2	<code>pn_fir_tau_flash</code>	BF16+TF32	1,797,638	653.54	546.78	516.11	4.52
suite 2s	<code>softmax</code>	BF16+TF32	10,795,008	441.16	411.03	379.83	— [†]
suite 2s	<code>sm_fir</code>	BF16+TF32	10,795,008	416.71	390.09	359.48	— [†]
suite 2s	<code>pn_fir_tau</code> (+6 τ)	BF16+TF32	10,795,014	428.52	409.33	377.36	— [†]
suite 3	<code>softmax</code>	BF16+TF32	1,797,632	77.93 ± 0.41	124.37 ± 0.26	117.13 ± 0.04	—
suite 3	<code>pn_fir_tau</code>	BF16+TF32	1,797,638	78.30 ± 0.45	122.15 ± 0.72	114.80 ± 0.49	—

Precision-controlled 50k, 3 seeds. Both suite 3 runs use BF16+TF32 across 3 seeds per variant (softmax: one seed-42 run plus seeds 43–44; pn: one seed-42 run plus seeds 43–44; `results/exp4d_softmax_50k_bf16.json`, `results/exp4e_multiseed_50k.json`), giving a matched 50k gap of -2.22 ± 0.46 val / -2.33 ± 0.49 test PPL for `pn_fir_tau`, holding on all 3 seeds (per-seed val gaps $-2.26/-2.76/-1.64$).

suite 2s notes. [†]Effective wall-clock per step including eval amortization (725/1121/789 ms for softmax/sm_fir/pn_fir_tau). A pure-step bench was attempted and abandoned: under WSL GPU-PV paging the FP32 suite crawls and eager `pn_fir_tau` OOMs the bench harness on backward. [‡]Peak VRAM under WSL paging is non-comparable (measured 14.1/14.2/18.2 GB on a 12 GB card) and omitted; a native-Linux measurement is left for future work.

Table 8: Quantization Hygiene: INT8 Activation Noise, Attention-Sink Mass, and Residual Outliers (Exp 4h). Per-token symmetric INT8 quantize-dequantize on every `nn.Linear` input (activation-only emulation, weights kept in fp), evaluated on the trained checkpoints. Sink mass is the mean attention mass on position 0 over the six layers. Source: `results/exp4h_int8_quant.json`.

Model	FP test PPL	INT8-sim PPL	Degrad.	Sink mass pos-0	Resid. absmax	Resid. kurtosis
D=64 softmax 50k	117.15	117.27	+0.10%	0.0177	5.03	4.17
D=64 pn_fir_tau 50k	114.25	114.35	+0.09%	0.0031	4.08	4.22
D=256 softmax 2.4k	379.81	380.96	+0.30%	0.0122	207.3	85.6
D=256 sm_fir 2.4k	359.47	360.75	+0.36%	0.0191	219.7	119.7
D=256 pn_fir_tau 2.4k	377.37	378.88	+0.40%	0.0013	176.5	78.5

Quantization hygiene (Exp 4h). Per-token symmetric INT8 quantize-dequantize on every `nn.Linear` input (Table 8) costs $< 0.5\%$ test PPL for every variant (D=64: 117.27 vs. 117.15 softmax, 114.35 vs. 114.25 pn; D=256: +0.30/+0.36/+0.40%) — no Softpick-scale low-bit win replicates at 2–11M parameters, and none is claimed. No massive sinks appear anywhere (max pos-0 mass 1.9%). The rectified hygiene advantage is real but conditional: `pn_fir_tau` carries 6–15 $\times$ less pos-0 mass (0.0031 vs. 0.0177 at D=64; 0.0013 vs. 0.0122–0.0191 at D=256, decaying to 0.0001 with depth) and 15–44% lower kurtosis across roles under D=256 scale-up stress, closing after converged training at D=64.

Table 9: Isolated 6-Layer MHA Stack and 1-Layer Kernel (`kern1`) Scaling ($D = 64, H = 4, d_k = 16$, uniform $B_M = B_N = 64$, Forward + Backward in BF16+TF32, 200 steps after 25 warmup, RTX 3060). Source: `results/exp4b_mha6_instrumentation.json`.

Seq, Batch	softmax	sm_fir	pn Eager	pn_triton	pn_flash	
$N = 256, B = 64$	34.26 ± 0.69	35.23 ± 0.80	101.02 ± 1.29	21.74 ± 4.97	17.66 ± 2.28	$[1.19 \pm 0.28]$
$N = 512, B = 64$	125.41 ± 2.34	127.52 ± 2.04	390.64 ± 6.12	37.76 ± 0.94	20.60 ± 2.73	$[2.05 \pm 0.14]$
$N = 1024, B = 64$	OOM	OOM	OOM	132.23 ± 1.01	46.77 ± 0.54	$[6.17 \pm 0.20]$
$N = 1024, B = 16$	126.35 ± 0.67	127.76 ± 0.34	393.42 ± 0.43	34.33 ± 0.52	18.13 ± 2.44	$[1.98 \pm 0.17]$

(a) step time (ms/step, fwd+bwd); (b) peak VRAM (MB). Per-row speedups ($4.65\times$ – $21.70\times$) are reported in §3.5.

Seq, Batch	softmax	sm_fir	pn Eager (MB)	pn_triton (MB)	pn_flash (MB)
$N = 256, B = 64$	815.2	827.5	1886.9	503.3	109.3
$N = 512, B = 64$	3086.3	3110.6	7341.4	1821.8	201.8
$N = 1024, B = 64$	OOM (> 12 GB)	OOM (> 12 GB)	OOM (> 12 GB)	6954.8	386.8
$N = 1024, B = 16$	3028.8	3040.9	7268.3	1751.3	109.3

Table 10: Silent-Tile Skipping Profile ($N = 1024, B = 16, B_M = B_N = 64$, Forward + Backward, 200 steps). While the isolated 1-layer kernel (`kern1`) speeds up by $1.73\times$ – $1.76\times$ when whole 64×64 tiles are silent ($\tau \geq +1.0$), scattered row silence ($\tau = +0.5, 69.1\%$ silent rows) leaves at least one positive entry in almost every 64×64 tile. The 6-layer module is launch-serialised (≈ 372 CUDA kernel launches per step, invariant across τ and $B \in \{4, 16\}$; $\approx 52\mu\text{s}$ per launch under WSL2 GPU-PV), so isolated-kernel savings do not translate into stack wall time; the `mha6` column here uses the original sequential protocol and its end-to-end difference is identified as a block-order artifact by the drift-controlled re-measurement (`experiments/exp4b_mha6_instrumentation.py`).

Threshold τ	Silent rows	Mean row mass	1L kernel (ms)	1L speedup	6L module (ms)	6L speedup
−0.5	0.0%	0.655	1.99 ± 0.14	1.00 $\times$	19.25 ± 3.35	1.00 $\times$
0.0	0.3%	0.469	1.94 ± 0.15	1.02 $\times$	18.77 ± 2.90	1.03 $\times$
+0.5	69.1%	0.048	1.91 ± 0.12	1.04 $\times$	18.24 ± 2.60	1.06 $\times$
+1.0	99.9%	0.0001	1.14 ± 0.21	1.73$\times$	18.32 ± 3.47	1.05 $\times$
+1.5	100.0%	0.0000	1.13 ± 0.20	1.75$\times$	18.07 ± 3.11	1.07 $\times$
+2.5	100.0%	0.0000	1.13 ± 0.18	1.76$\times$	17.65 ± 2.32	1.09 $\times$

Drift-controlled re-measurement of the 6-layer stack. To separate the kernel-level saving from the stack-level wall, we re-instrumented the exact configuration of Table 10 ($N=1024$, $B=16$, BF16 autocast, matmul TF32 off, seed 42) with τ blocks interleaved in rotating order (three ($\tau = -0.5, \tau = +2.5$) pairs, three low-start and three high-start; 25 warmup + 120 timed steps per block), CUDA-event spans around the forward residual loop and the backward pass, per-layer per-tile skip diagnostics, `torch.profiler` launch counts, and a $B \in \{16, 4\}$ workload-scaling discriminator. Table 11 reports the outcome: the stack wall is τ -flat (-0.43 ± 0.43 ms, $|z| \approx 1.0$, $n = 6$ block pairs) even though the isolated kernel is not ($2.167 \pm 0.027 \rightarrow 1.464 \pm 0.032$ ms, $1.48\times$, i.e. 0.703 ± 0.042 ms per layer, 4.2 ms over six layers), the number of CUDA kernel launches per step is invariant across τ and B (372.7 vs. 372.3), and 100.0% of the 64×64 tiles are fully silent in every layer at $\tau = +2.5$ against the 46.9% causal floor at $\tau = -0.5$. The launch-bound diagnosis is then confirmed by intervention: capturing the entire 6-layer forward+backward in a CUDA graph (Table 12) amortises dispatch to a single replay and surfaces the saving in full ($12.10 \pm 0.03 \rightarrow 4.91 \pm 0.02$ ms, $\Delta = -7.20 \pm 0.02$ ms), while the eager walls measured in the same process stay τ -flat (+2.04 ms, within drift). The design lever at this stack scale is therefore fewer launches per step (graph capture or kernel fusion), not more skipping.

Table 11: Drift-Controlled Re-Measurement of the 6-Layer Stack (Exp 4b). The τ blocks are interleaved in rotating order (three ($\tau = -0.5, \tau = +2.5$) pairs, three low-start and three high-start; 25 warmup + 120 timed steps per block) at $N=1024$, $B=16$, BF16 autocast, matmul TF32 off, seed 42, so that the wall time is measured against within-session drift instead of against the block order. Wall times, spans and kernel times are in milliseconds; the last row is the $B\ 16 \rightarrow 4$ workload-scaling comparison (standard errors over the two blocks: $\tau = -0.5$: 19.50 ± 0.20 vs. 18.88 ± 0.16 ms; $\tau = +2.5$: 18.49 ± 0.39 vs. 19.76 ± 0.36 ms). Source: `results/exp4b_mha6_instrumentation.json`.

Quantity	$\tau = -0.5$ (active)	$\tau = +2.5$ (silent)	Δ
Stack wall per step (6 blocks)	20.30 ± 0.30	19.87 ± 0.33	-0.43 ± 0.43 ($ z \approx 1.0$)
Forward residual-loop span	8.23	8.37	+0.15 (τ -invariant)
Backward span	11.61	10.96	-0.65 (vs. 4.2 predicted)
Isolated kernel wall (kern1)	2.167 ± 0.027	1.464 ± 0.032	-0.703 ± 0.042 ($1.48\times$)
Fully-silent 64×64 tiles/layer	46.9%	100.0%	+53.1 p.p.
CUDA kernel launches per step	372.7	372.3	invariant ($\approx 52 \mu\text{s}/\text{launch}$)
Stack wall, $4\times$ less GPU work	19.50 vs. 18.88	18.49 vs. 19.76	$3.08\times$ / $3.23\times$ SEM

Table 12: Launch Amortisation Isolated by CUDA-Graph Replay (Exp 4c). The whole 6-layer forward+backward is captured in a CUDA graph, so per-step launch dispatch is replaced by a single replay; the eager walls are measured in the same process. The replay-determinism check gives a maximum relative gradient deviation of 5.5×10^{-3} at $\tau = -0.5$ and exactly 0 at $\tau = +2.5$. Source: `results/exp4c_cuda_graphs.json`.

Quantity (rotated interleave)	$\tau = -0.5$ (fully active)	$\tau = +2.5$ (all tiles silent)	Δ (+2.5 vs. -0.5)
Eager stack wall (launch-bound)	21.46 ± 0.47 ms	23.50 ± 0.68 ms	+2.04 ms (within drift)
CUDA-graph replay wall (fwd + bwd)	12.10 ± 0.03 ms	4.91 ± 0.01 ms	-7.20 ± 0.02 ms
Speed-up vs. eager at the same τ	$1.77\times$	$4.79\times$	—

Supporting figures for Experiment 4. Figures 10 and 11 show the 50k-step training curves and the per-seed perplexity of the matched precision-controlled protocol. Figure 12 reports the per-layer silence diagnostics behind Table 10, and Figure 13 illustrates the silent-tile skipping mechanism. Figure 14 compares the saving predicted from the single-kernel measurement against the saving realized by the 6-layer stack, Figure 15 isolates launch amortisation by CUDA-graph replay, and

Figure 16 quantifies the measurement-order drift. Figure 17 reports the INT8 outlier and attention-sink diagnostics of Experiment 4h.

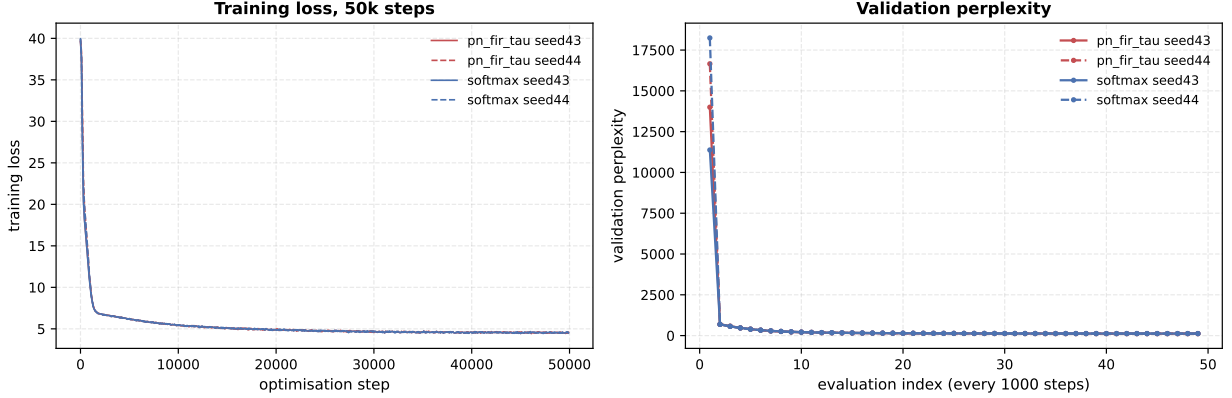


Figure 10: Training history of the 50k-step runs. Left: training loss against optimisation step. Right: validation perplexity at the periodic evaluations. Curves are $n = 2$ seeds per variant (seeds 43 and 44); no seed-level error band is shown, because two seeds do not support one. Source: results/exp4e_multiseed_50k.json (runs.<variant>/<seed>.history).

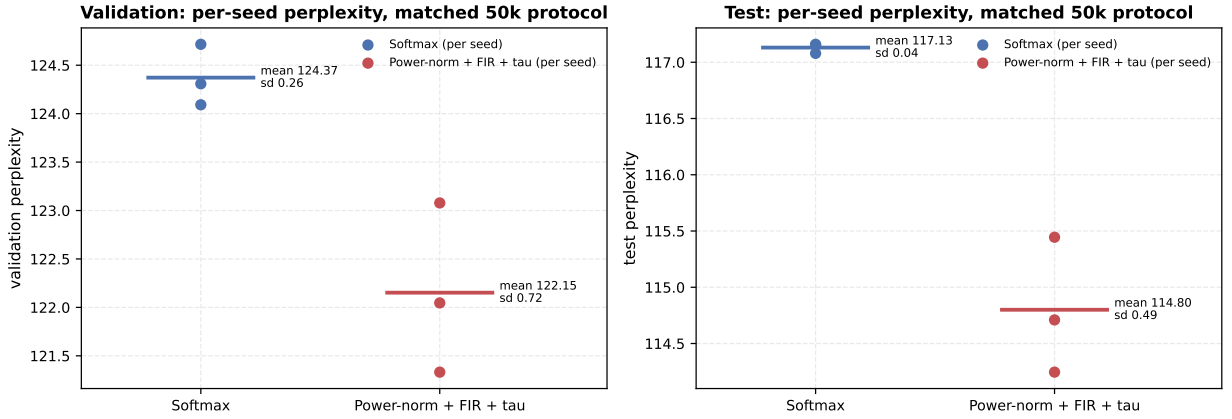


Figure 11: Per-seed perplexity under the matched 50k protocol. Each dot is one seed ($n = 3$ per variant); the bar is the mean over those seeds and the annotation gives the mean and the standard deviation over seeds. The seed-42 point of the rectified variant is the reference run of the matched-precision record; the two remaining seeds come from the multi-seed run. Source: results/exp4e_multiseed_50k.json and results/exp4d_softmax_50k_bf16.json.

4 Limitations and Negative Results

Six limitations bound the claims above.

1. **Scale:** all multi-seed language-modeling results are at $D=64$ (1.8M parameters, $L = 6$, seq=256); a single-seed scale-out at $D=256/N=512$ (10.8M, suite 2s: `sm_fir` 359.5 test, `pn_fir_tau` 377.4, `softmax` 379.8) shows no collapse and preserves the ranking, with a replicated tau profile (layer 0 at -0.50). An auxiliary GPT-2 124M retrofit probe (Softmax→`pn_fir_tau` weight transplant, 20k fine-tune steps) reached Test PPL 47.05 vs. 21.43 for a Softmax retrofit of the same budget; its raw artifacts are not distributed with this work. We attribute the gap to co-adaptation: pre-trained W_Q, W_K are optimized for the

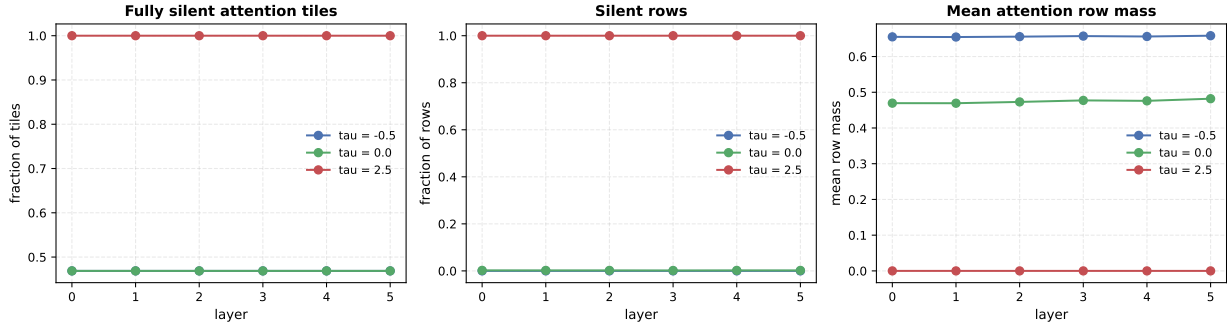


Figure 12: Per-layer silence diagnostics of the 6-layer stack for three thresholds. Left: fraction of fully silent attention tiles. Middle: fraction of silent rows. Right: mean attention row mass. Single instrumented run per threshold (one run per τ), so no seed spread is shown. Source: `results/exp4b_mha6_instrumentation.json` (`diagnostics_per_tau`).

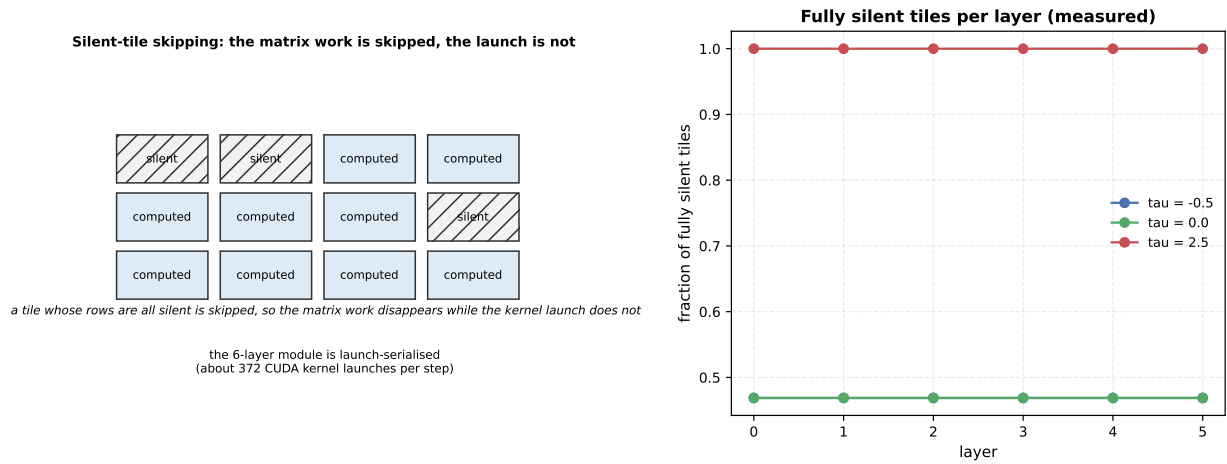


Figure 13: Schematic of silent-tile skipping. Left: a tile grid in which tiles whose rows are all silent are skipped, so the matrix work disappears while the kernel launch does not; the 6-layer module is launch-serialised with about 372 CUDA kernel launches per step. Right: the measured fraction of fully silent tiles per layer and threshold. Single instrumented run per threshold. Source: `results/exp4b_mha6_instrumentation.json` (`diagnostics_per_tau`) and `src/triton_pnfirtau.py`.

translation-invariant geometry of the Softmax simplex, whereas the FIR-induced score shift changes that geometry; a 20k fine-tune cannot re-learn the projection matrices, so the gap is expected. Continuous pre-training or low-rank/evolutionary adaptation is the appropriate protocol at this scale; no GPT-2-class result is claimed. At 1.8M parameters attention sinks are present but modest (Exp 4h: max position-0 mass 1.9%), so sink-suppression benefits are bounded at this scale. RAG-style behavior is evaluated on a controlled synthetic task, not on a production retrieval corpus.

- Matched 50k with variance, small scale:** Table 7 (suite 3) is precision-controlled across 3 seeds per variant, with a -2.22 ± 0.46 val / -2.33 ± 0.49 test PPL gap for `pn_fir_tau`, holding on all 3 seeds; pn shows larger seed spread than softmax (val std 0.72 vs. 0.26), and all results are at $D=64$ with no larger-model replication. All quoted throughput numbers are single-seed.
- Tile-granularity and launch serialisation:** silent-tile skipping accelerates the isolated

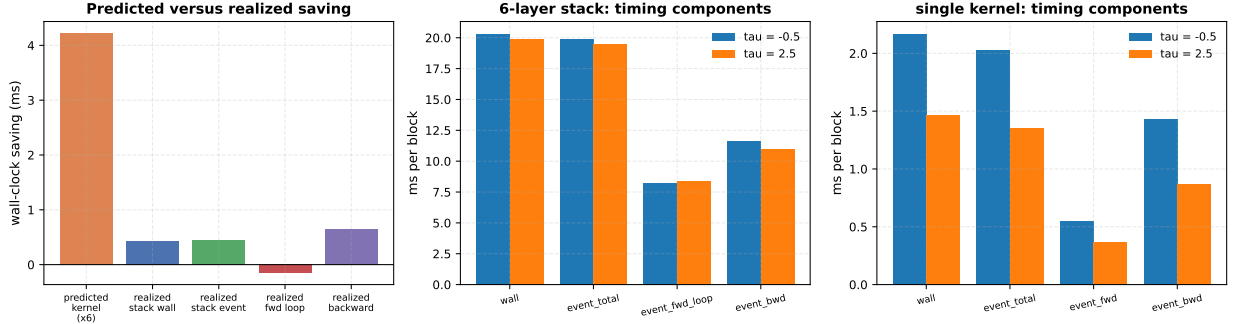


Figure 14: Predicted versus realized wall-clock saving. Left: the saving predicted from the single-kernel measurement against the saving actually realized by the 6-layer stack and its forward/backward split. Centre and right: timing components of the stack and of the single kernel. Single instrumented run per threshold; the error bars of the underlying block timings are reported in the text, not drawn here. Source: `results/exp4b_mha6_instrumentation.json` (attribution, stack_timing, kern1_timing).

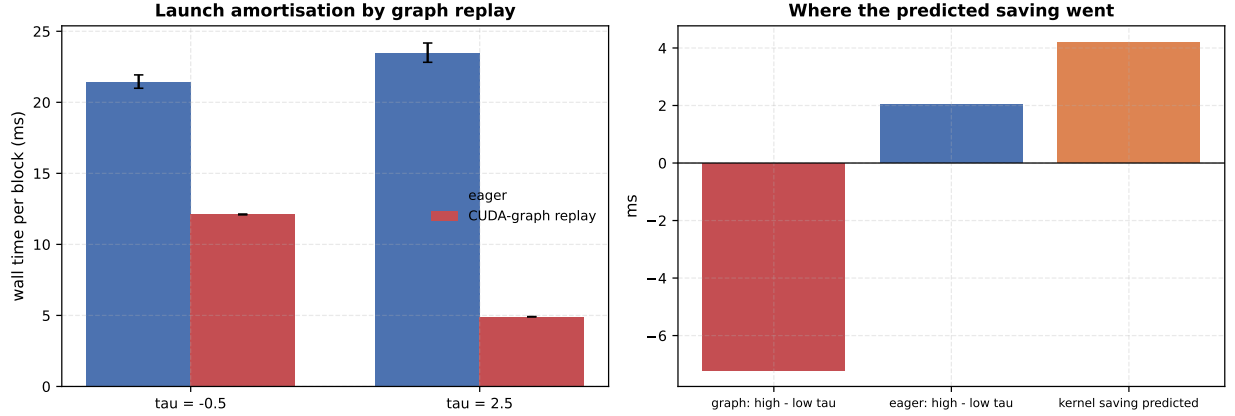


Figure 15: Launch amortisation isolated by CUDA-graph replay. Left: wall time per block of the 6-layer stack under eager execution and under graph replay, for two thresholds; error bars are the standard error over the measured blocks of a single run. Right: the high-minus-low threshold difference under graph replay and under eager execution, against the saving predicted from the kernel measurement. Source: `results/exp4c_cuda_graphs.json` (combined, attribution).

kernel ($1.48\times$ re-measured, $1.76\times$ original) but is launch-serialised in the 6-layer stack (≈ 372 CUDA launches per step, invariant across τ and $B \in \{4, 16\}$; wall unchanged at $4\times$ less GPU work under WSL2 GPU-PV). The end-to-end stack saving attributed to tile skipping (10, mha6 column) was reproduced and falsified as a block-order artifact by the drift-controlled re-measurement (exp4b). A CUDA-graph replay confirms the mechanism by intervention: amortising launches surfaces the saving fully ($12.10 \pm 0.03 \rightarrow 4.91 \pm 0.02$ ms, $\Delta = -7.20 \pm 0.02$ ms, exp4c).

- Decoy flooding is memory-length-specific:** exact copies of the queried key at high multiplicity ($n_d \geq 4$; 56.9% at $n_d=8$, $\epsilon=0.2$ under the 7-tap FIR) do not defeat short-memory linear filtering: a 2-tap shift holds $99.9\% \pm 0.2\%$ (sm_shiftmu) and $92.7\% \pm 10.4\%$ (pn_shiftmu_tau) at $n_d=8$ (Exp 6), so the failure is specific to the 7-tap memory length, at the cost of seed-dependent τ - μ equilibria in the rectified branch.
- τ calibration regime:** under positive-only training, gradients on τ come from active rows

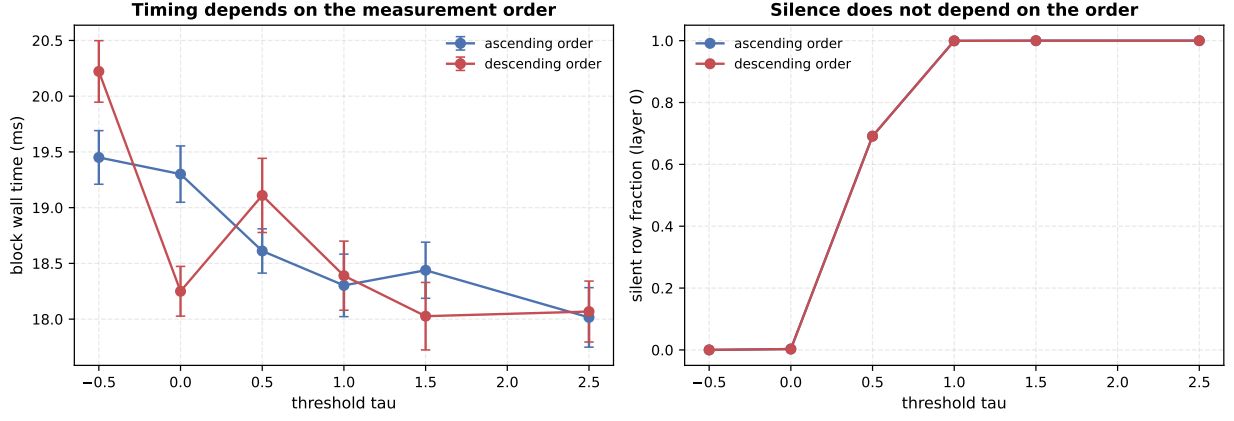


Figure 16: Sensitivity of the block timing to the order in which the thresholds are measured. Left: block wall time against threshold, one line per measurement order; error bars are the standard error over the blocks of a single run. Right: the silent-row fraction of the first layer, which does not depend on the order. Source: `results/exp4b_sequential_repro.json` (runs).

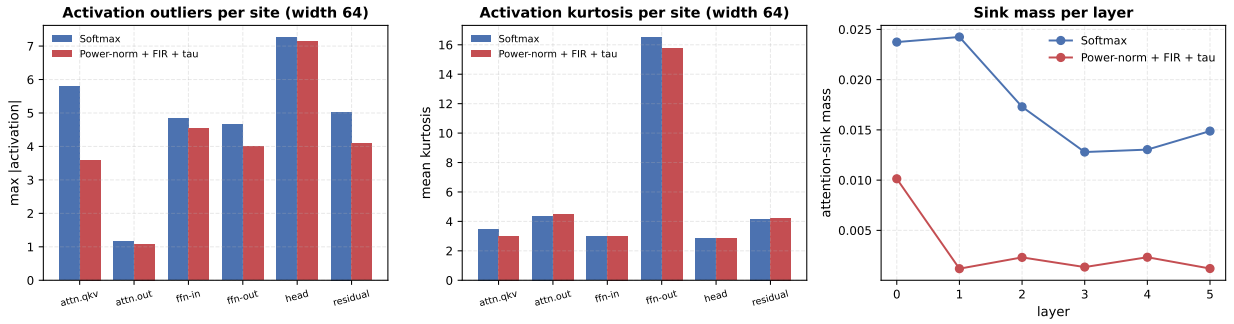


Figure 17: Activation outliers and attention-sink mass of the width-64 runs. Left: maximum absolute activation per site. Middle: mean kurtosis per site. Right: sink mass per layer. One trained checkpoint per variant (no seed spread available). Source: `results/exp4h_int8_quant.json` (`runs.<run_id>.outliers`, `runs.<run_id>.sink`).

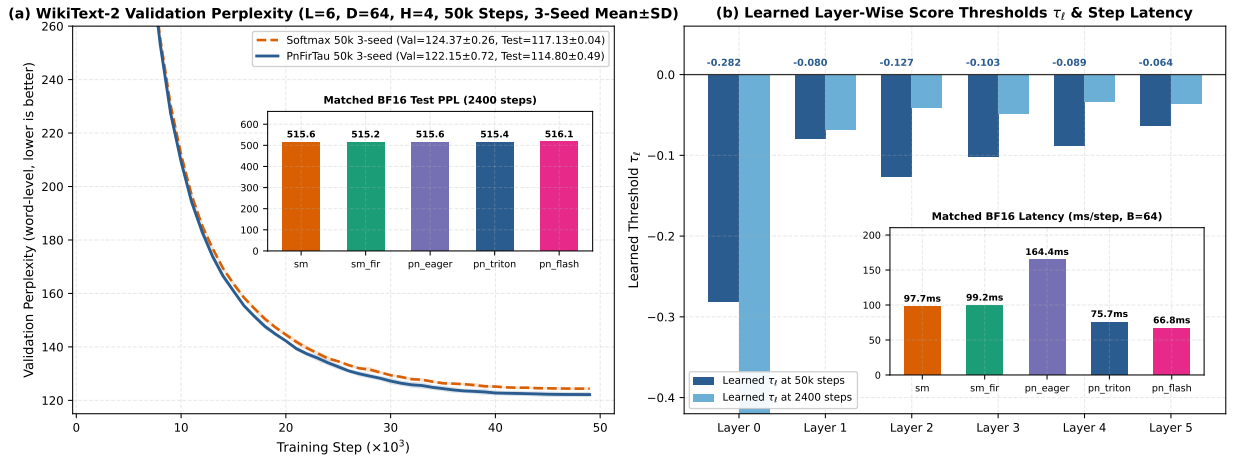


Figure 18: WikiText-2 50,000-step validation perplexity trajectory (3-seed mean±SD bands, suite 3 finals in legend) and 2,400-step matched BF16 inset (left), and learned layer-wise thresholds τ_ℓ with matched BF16 step-time inset across all 5 variants (right). Source: `results/exp4_wikitext2_lm.json`, `results/exp4d_softmax_50k_bf16.json`, `results/exp4e_multiseed_50k.json`.

only, so its resting point against pure-noise rows varies across seeds unless the post-warmup absent-key penalty (`pn_fir_tau_reg`) is applied.

6. **Hyperparameter sensitivity:** $p = 10$ and $\sigma = 2.0$ are fixed a priori; a sensitivity sweep over these values is left for future work.

5 Conclusion and Practical Guide for Transformer Designers

We have shown that causal FIR temporal smoothing strictly on K enables single-layer content-addressable value-slot binding ($W_{\text{query},j^*+1} = 0.729\text{--}0.792$, lifting clean recall from 35.8%–44.5% to $100.0\% \pm 0.0\%$ on Exp 1 and Exp 2), established mass constancy along uniform shift rays (Proposition 1), explained why uncalibrated `pn_fir` suffers score-shift starvation on positive-only training while simplex normalizers (`Softmax`, `Sparsemax`, `Entmax`) fail residual preservation on unanswerable queries (27.2%–34.4% hallucination rate in Exp 5) and row-mean centering (`pn_fir_c`) leaks $\approx 124\times$ more noise ($\sum W_{\text{unans}} = 0.265$), and demonstrated that `pn_fir_tau` achieves $100.0\% \pm 0.0\%$ residual preservation (0.0% hallucination rate, $\sum W_{\text{unans}} = 0.002 \pm 0.002$, $50\times$ lower continuous residual leakage than `Softmax_1` and `Quiet Softmax`) under pure Cross-Entropy loss while matching and slightly improving 6-layer WikiText-2 perplexity at 1.8M parameters.

For practitioners building and deploying Transformer models, our findings translate into two actionable architectural takeaways:

1. **Immediate Zero-Parameter Single-Layer Induction (`sm_fir`):** If the primary goal is to equip a standard causal Transformer with single-layer key-value induction binding (compressing two-layer induction heads into one layer) without custom attention kernels, applying the fixed 7-tap causal Gaussian FIR filter strictly to the key tensor K (`sm_fir`) requires zero learnable parameters, adds only +1.5% step-time overhead in BF16 (+2.0% in FP32), and can be paired with `triton_causal_fir` for zero-copy strided filtering.
2. **Hallucination-Free Residual Preservation, Sink-Free Attention, and Transcendental-Free Hardware Acceleration (`pn_fir_tau_triton` / `pn_fir_tau_flash`):** If the architecture requires attention heads to **abstain cleanly when queried information is absent from the context**—such as RAG over noisy passages, memory-augmented Transformers, or models designed for low-bit INT4/INT8 quantization and streaming KV caches without BOS Attention-Sink outliers—replacing simplex normalizers with sub-simplex rectified power-normalization (`pn_fir_tau`) eliminates context-induced residual hallucinations (0.0% vs. 27.2%–34.4% for `Softmax`/`Sparsemax`/`Entmax`) and suppresses continuous absent-key residual leakage $50\times$ more tightly than virtual-sink `Softmax`. Moreover, when deployed via our fused Triton kernels (`PnFirTauTritonMHA` and `PnFirTauFlashMHA` in `src/triton_pnfirtau.py`), evaluating $(S'_{t,j})_+^{10}$ in 3 register multiplications with zero `exp()` calls and zero online tile rescaling reduces full-LM BF16 step time by 22.5%–31.6% **below standard Softmax** ($2.17 \times \text{--} 2.46\times$ faster than eager PyTorch) and achieves $21.70\times$ **speedup** ($6.97\times$ vs. **softmax**) and $66.5\times$ **VRAM reduction** on a 6-layer MHA stack at $N = 1024, B = 16$.

Declarations

Competing interests. The author declares no competing interests.

Funding. This research received no external funding.

Data and code availability. The reproducibility package (source code, experiment suites, raw JSON results, and figure-generation pipeline) is provided with this preprint; an archived copy will be deposited on Zenodo upon publication.

Author contributions. M.Z. conceived the study, implemented the method and kernels, ran the experiments, analysed the results, and wrote the manuscript.

Ethics. Not applicable; this study uses no human or animal subjects.

References

- [1] B. Zhang, I. Titov, and R. Sennrich, “Sparse attention with linear units,” *arXiv:2104.07012*, 2021.
- [2] Z. M. K. Zuhri, E. H. Fuadi, and A. F. Aji, “Softpick: No attention sink, no massive activations with rectified softmax,” *arXiv:2504.20966*, 2025.
- [3] R. Z. Wang, “Abstention and noise filtering: Two missing primitives of softmax attention,” *arXiv:2609.22005*, 2026.
- [4] M. Sun, X. Chen, J. Z. Kolter, and Z. Liu, “Massive activations in large language models,” *arXiv:2402.17762*, 2024.
- [5] B. Peng, E. Alcaide, Q. Anthony, et al., “RWKV: Reinventing RNNs for the transformer era,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 14048–14077, doi:10.18653/v1/2023.findings-emnlp.936 (preprint: *arXiv:2305.13048*).
- [6] C. Olsson, N. Elhage, N. Nanda, et al., “In-context learning and induction heads,” *Transformer Circuits Thread*, 2022, *arXiv:2209.11895*.
- [7] A. Vaswani, N. Shazeer, N. Parmar, et al., “Attention is all you need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, *arXiv:1706.03762*.
- [8] D. R. So, W. Mañke, H. Liu, Z. Dai, N. Shazeer, and Q. V. Le, “Primer: Searching for efficient transformers for language modeling,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, *arXiv:2109.08668*.
- [9] A. F. T. Martins and R. F. Astudillo, “From softmax to sparsemax: A sparse model of attention and multi-label classification,” in *International Conference on Machine Learning (ICML)*, 2016, *arXiv:1602.02068*.

- [10] B. Peters, V. Niculae, and A. F. T. Martins, “Sparse sequence-to-sequence models,” in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 1504–1519, doi:10.18653/v1/P19-1146 (preprint: *arXiv:1905.05702*).
- [11] G. M. Correia, V. Niculae, and A. F. T. Martins, “Adaptively sparse transformers,” in *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2174–2184, doi:10.18653/v1/D19-1223 (preprint: *arXiv:1909.00015*).
- [12] E. Miller, “Attention is off by one,” online post, July 24, 2023. [Online]. Available: <https://www.evanmiller.org/attention-is-off-by-one.html>. [Accessed: October 19, 2026].
- [13] Y. Bondarenko, M. Nagel, and T. Blankevoort, “Quantizable transformers: Removing outliers by helping attention heads do nothing,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, *arXiv:2306.12929*.
- [14] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis, “Efficient streaming language models with attention sinks,” in *International Conference on Learning Representations (ICLR)*, 2024, *arXiv:2309.17453*.
- [15] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, “FlashAttention: Fast and memory-efficient exact attention with IO-awareness,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, *arXiv:2205.14135*.