THE ENTROPY BIAS OF HYBRID ATTENTION: TRAINING MODELS TO BE SPARSE-READY

Anonymous authors
Paper under double-blind review

ABSTRACT

Hybrid language models improve long-context efficiency by concentrating global information exchange into a small number of full-attention layers. We find that these layers exhibit a systematic entropy bias: compared with full-attention Transformers, their attention is more diffuse, leaving less mass recoverable under a fixed sparse budget and making them less robust to sparsification. Importantly, this behavior is not an inherent limitation of hybrid architectures, but an optimization outcome that can be shaped during training. Through controlled pretraining experiments, we show that an architecture-aware concentration prior reduces entropy and substantially improves sparse long-context performance. The gains extend across hybrid architectures, model sizes, and longer contexts, and remain evident with learned QK gains and trained sparse indexers. Entropy saturation further offers a practical guide to prior strength. Together, these findings establish architecture-aware attention concentration as a key training principle for sparse-ready hybrid models.

1 INTRODUCTION

Agentic language models have made substantial progress in automating complex workflows, yet they must process long codebases, retrieved documents, and accumulated tool interactions. At these lengths, full softmax attention incurs quadratic prefill work and a key–value cache that grows with the context in every layer. Hybrid architectures reduce this cost by interleaving efficient attention with a smaller number of full-attention layers: the former handle local or recurrent processing, while the latter retain direct access to distant tokens (Qiao et al., 2026). This division of labor has enabled efficient long-context modeling in systems such as Qwen3.5 (Qwen Team, 2026a), Kimi K3 (Kimi Team, 2026), and MiMo-V2-Flash (Xiao et al., 2026). To push efficiency further under demanding serving workloads, recent models such as Qwen3.8-Flash-Next (Qwen Team, 2026b) and GLM-5.3-Flash (Z.ai, 2026) sparsify the remaining global-attention layers. Existing sparse-attention methods have advanced token and block selection, indexing, and sparse training (DeepSeek-AI, 2025; Lu et al., 2025; Zhao et al., 2025; Zan et al., 2026). Yet it remains unclear whether a full-attention layer in a hybrid model is as tolerant of sparsification as its counterpart in a full-attention model.

Attention entropy offers a way to study this question. Earlier work linked abruptly collapsing attention entropy to unstable Transformer training and introduced logit-control mechanisms such as σReparam and query–key normalization (Zhai et al., 2023; Henry et al., 2020). In a different regime, the growing number of candidate keys can dilute attention to salient tokens; length-dependent logit scaling has therefore been studied to preserve concentration when extrapolating to longer contexts (Chiang & Cholak, 2022; Su, 2021; Nakanishi, 2025). These studies show that concentration matters, but address training stability or context-length extrapolation rather than the sparse readiness of different architectures.

In this work, we first find that the full-attention layers of hybrid architectures develop systematically higher entropy and retain less attention mass in their top-k keys than those of a full-attention model. This *entropy bias* is broadly distributed across layers and accompanies a wider allocation of long-range retrieval across heads. It does not imply that a hybrid is necessarily worse under dense attention. Instead, it can make the remaining global layers less robust to sparsification, with different hybrid components exhibiting different degrees of robustness. Motivated by the connection between

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.

retained mass and sparse-approximation error, we study an architecture-aware *concentration prior*: a fixed logit scale applied to the hybrid's full-attention layers during pretraining.

Controlled pretraining ablations show that this prior lowers entropy, raises top-k retained mass, and substantially improves long-context performance under sparse attention. The effect holds for both hybrid components, persists across model scales, and remains useful after context extension. We also find that entropy saturates as the scale increases, which offers a practical guide to choosing a scale that balances sparse long-context performance and general ability. Additional ablations favor applying the prior to full-attention layers and introducing it early; learned QK gains alone do not remove its benefit. A similar benefit appears in indexer-based sparse training following DeepSeek Sparse Attention (DeepSeek-AI, 2025). Together, our findings reveal that the sparse readiness of hybrid architectures is shaped by a systematic entropy bias, and show that an architecture-aware concentration prior applied during pretraining can mitigate this bias. This offers a practical principle for training efficient frontier models.

2 PRELIMINARIES

2.1 HYBRID MODELS

Full Attention. Let $\boldsymbol{q}_t, \boldsymbol{k}_t \in \mathbb{R}^{d_k}$ and $\boldsymbol{v}_t \in \mathbb{R}^{d_v}$ denote the query, key, and value vectors of one head at position t, respectively. Given a visible token set $\mathcal{C}_t$, causal softmax attention (Vaswani et al., 2017) computes

$$\boldsymbol{o}_t = \sum_{s \in \mathcal{C}_t} a_{t,s} \boldsymbol{v}_s, \qquad a_{t,s} = \frac{\exp\big(\boldsymbol{q}_t^\top \boldsymbol{k}_s / \sqrt{d_k}\big)}{\sum_{j \in \mathcal{C}_t} \exp\big(\boldsymbol{q}_t^\top \boldsymbol{k}_j / \sqrt{d_k}\big)}. \tag{1}$$

Full attention uses $\mathcal{C}_t = \{1, \ldots, t\}$, giving each query direct access to all preceding tokens with $\mathcal{O}(T^2)$ pairwise interactions for a sequence of length T.

Sliding Window Attention. Sliding-window attention (SWA) restricts the visible tokens in Equation 1 to $\mathcal{C}_t = \{\max(1, t - w + 1), \ldots, t\}$ for window size w, reducing the sequence-length complexity to $\mathcal{O}(Tw)$ (Beltagy et al., 2020).

Linear Attention. Linear attention and related recurrent mixers (Qin et al., 2024; Dao & Gu, 2024) avoid explicit pairwise comparisons with the entire prefix by compressing past information into a fixed-size state. For fixed state dimensions, this gives $\mathcal{O}(T)$ sequence-length cost. Gated DeltaNet (GDN) is a representative variant that updates the state with a gated delta rule (Yang et al., 2025). In a canonical single-head form, with $\boldsymbol{S}_t \in \mathbb{R}^{d_v \times d_k}$, data-dependent gates $\alpha_t, \beta_t \in (0, 1)$, and an ℓ_2-normalized key $\boldsymbol{k}_t$, it computes

$$\boldsymbol{S}_t = \alpha_t \boldsymbol{S}_{t-1}\big(\boldsymbol{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top\big) + \beta_t \boldsymbol{v}_t \boldsymbol{k}_t^\top, \qquad \boldsymbol{o}_t = \boldsymbol{S}_t \boldsymbol{q}_t. \tag{2}$$

The delta rule removes the association stored along $\boldsymbol{k}_t$ before writing the new key–value association, while α_t controls forgetting.

A hybrid model interleaves full-attention layers with efficient-attention components such as SWA or GDN. This design encourages the efficient layers to handle local language modeling or recurrent updates, while the full-attention layers retain direct access to long-range information (Qiao et al., 2026). It thereby reduces long-context cost without removing global token interactions entirely.

2.2 SPARSE ATTENTION

Sparse attention restricts the main attention operator to a selected subset $\mathcal{S}_t \subseteq \{1, \ldots, t\}$, with $|\mathcal{S}_t| \le k$. Writing $z_{t,s} = \boldsymbol{q}_t^\top \boldsymbol{k}_s / \sqrt{d_k}$, its output is

$$\boldsymbol{o}_t^{\mathrm{sp}} = \sum_{s \in \mathcal{S}_t} \frac{\exp(z_{t,s})}{\sum_{j \in \mathcal{S}_t} \exp(z_{t,j})} \boldsymbol{v}_s. \tag{3}$$

Once $\mathcal{S}_t$ is available, the main attention cost decreases from $\mathcal{O}(T^2)$ to $\mathcal{O}(Tk)$. The key challenge is to select a small subset that retains the important attention mass.

DeepSeek Sparse Attention (DSA) addresses this problem with a lightweight indexer followed by fine-grained top-k token selection (DeepSeek-AI, 2025):

$$I_{t,s} = \sum_{r=1}^{H_I} w_{t,r}^I \, \mathrm{ReLU}\big((\boldsymbol{q}_{t,r}^I)^\top \boldsymbol{k}_s^I\big), \qquad \mathcal{S}_t = \underset{s \le t}{\mathrm{TopK}} \, I_{t,s}. \tag{4}$$

Here, a small number of low-dimensional indexer heads estimate global token importance, and the main attention module operates only on the selected key–value entries. The indexer is trained to align with the dense attention distribution before the model adapts to the resulting sparse pattern.

To characterize attention concentration at a given sparse budget, let $\boldsymbol{a} \in \mathbb{R}^n$ be a dense attention distribution and let $\mathcal{S}_k$ contain the indices of its k largest entries. Its *retained mass* is

$$M_k(\boldsymbol{a}) = \sum_{i \in \mathcal{S}_k} a_i. \tag{5}$$

This oracle top-k set retains the largest possible probability mass among all subsets of k keys, providing a diagnostic independent of indexer quality.

2.3 Attention Entropy

Attention concentration. For an attention distribution over $\mathcal{C}_t$, we define its Shannon entropy and effective support size as

$$\mathcal{H}_t = -\sum_{s \in \mathcal{C}_t} a_{t,s} \log a_{t,s}, \qquad N_{\mathrm{eff},t} = \exp(\mathcal{H}_t). \tag{6}$$

Entropy measures how diffusely probability is distributed across visible tokens, ranging from 0 for one-hot attention to $\log |\mathcal{C}_t|$ for uniform attention. Effective support expresses this quantity as the number of equally weighted tokens with the same entropy.

Entropy control for training stability. Abrupt entropy collapse can accompany training instability (Zhai et al., 2023), motivating control of attention logits to avoid uncontrolled concentration. QKNorm (Henry et al., 2020) normalizes each query and key along the head dimension and replaces the usual $1/\sqrt{d_k}$ factor with a learned logit scale γ:

$$\tilde{\boldsymbol{q}}_t = \frac{\boldsymbol{q}_t}{\|\boldsymbol{q}_t\|_2}, \qquad \tilde{\boldsymbol{k}}_s = \frac{\boldsymbol{k}_s}{\|\boldsymbol{k}_s\|_2}, \qquad a_{t,s}^{\mathrm{QKNorm}} = \mathrm{softmax}_{s \in \mathcal{C}_t}\Big(\gamma \, \tilde{\boldsymbol{q}}_t^\top \tilde{\boldsymbol{k}}_s\Big). \tag{7}$$

This mitigates softmax saturation, while σReparam controls the spectral norms of projection matrices to prevent entropy collapse (Zhai et al., 2023). These approaches seek stable optimization rather than uniformly maximizing attention entropy. They do not examine whether controlling entropy improves robustness to sparsification.

Entropy control for length extrapolation. As the context grows beyond the training length, additional keys can dilute the attention assigned to salient tokens. Length-dependent logit scaling has therefore been studied to compensate for this dilution and approximately preserve attention concentration (Chiang & Cholak, 2022; Su, 2021; Nakanishi, 2025). One normalized log-n form is

$$a_{t,s}^{(\log n)} = \mathrm{softmax}_{s \in \mathcal{C}_t}\left(\frac{\log n}{\log n_0} \, z_{t,s}\right), \qquad n = |\mathcal{C}_t|, \tag{8}$$

where $n_0 > 1$ is a reference training length, so the original logit scale is recovered at $n = n_0$. Other formulations learn the coefficient of $\log n$. Together, these perspectives show that the desired concentration depends on the objective and context length. They motivate examining how architecture shapes attention concentration and how this affects sparsification.

3 The Entropy Bias of Hybrid Attention

3.1 Hybrid Full-Attention Layers Develop Higher Entropy

We compare 24-layer Full, SWA-hybrid, and GDN-hybrid models trained with the same pretraining pipeline; details are provided in Appendix A.

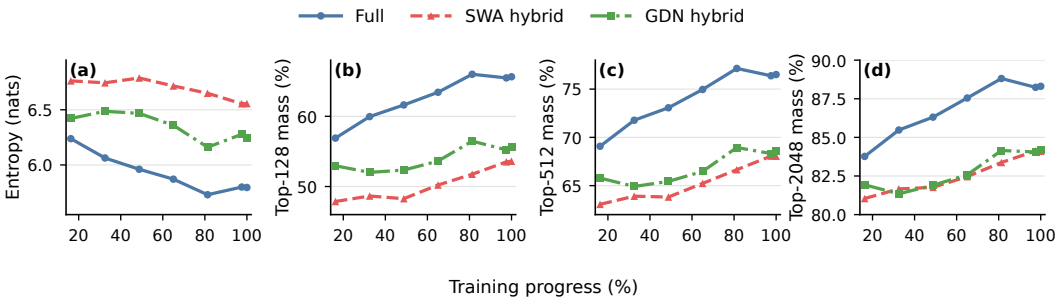

Figure 1: Attention entropy and top-k retained mass in full-attention layers throughout pretraining.

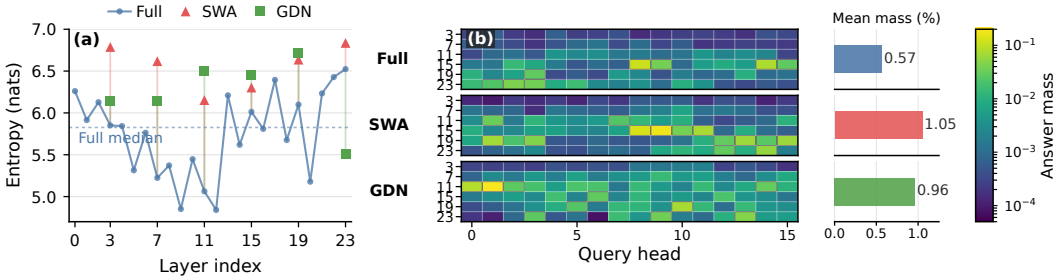

Figure 2: Layerwise entropy and retrieval: (a) entropy by layer; (b) mean answer-token attention mass by layer and head, with model averages.

Throughout training, we periodically measure the final prefill query of 50 fixed 16K RULER prompts, averaging over the 16 query heads and six matched full-attention layers. Figure 1 traces the aligned pretraining trajectories. Both hybrids consistently exhibit higher full-attention entropy than Full, reaching 6.56 nats for SWA and 6.24 for GDN at the end of training, compared with 5.80 for Full.

The higher entropy is accompanied by lower coverage under a fixed token budget: SWA and GDN retain 12.0 and 10.0 percentage points less mass than Full at $k = 128$, with gaps also present at larger budgets. Thus, even oracle top-k selection discards more mass in the hybrids. We refer to this systematic tendency as the *entropy bias* of hybrid attention.

3.2 Entropy Bias Accompanies Broader Global Retrieval

Figure 2(a) compares all 24 Full attention layers with every full-attention layer of the hybrids. All six SWA full-attention layers have higher entropy than their position-matched Full counterparts and exceed the median over all Full layers. The same holds for five of the six GDN layers, with the final layer as the exception. The entropy bias is thus broadly distributed rather than driven by a few outliers.

We also measure each head's mean answer-token attention mass on 400 matched 16K RULER needle-in-a-haystack prompts, comparing the same six layer positions and 96 heads in each model (Figure 2(b)). Mean answer mass increases from 0.57% in Full to 1.05% in SWA and 0.96% in GDN. The fraction of heads with answer mass above 0.02 rises from 8.3% to 14.6% and 10.4%, respectively, indicating more broadly distributed retrieval signal in both hybrids. Additional analyses are provided in Appendix C.

Hybrid full-attention layers therefore combine broadly distributed retrieval signal across heads with more diffuse attention across tokens.

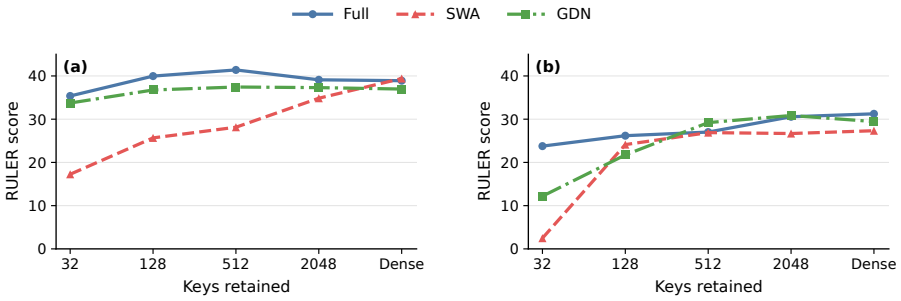

Figure 3: RULER scores under exact per-head top-k attention at (a) 16K and (b) 64K, including dense-attention baselines.

3.3 Hybrid Architectures Suffer More from Sparsification

We measure *sparse robustness* by comparing dense 16K and 64K RULER scores with exact per-head top-k selection in full-attention layers (Figure 3).

Full preserves substantially more long-context performance under aggressive sparsification, with the clearest gap at 64K. At top-32, Full loses only 7.47 points from its dense score, versus 17.26 for GDN and 24.85 for SWA; its remaining score is 23.77, compared with 12.19 and 2.48. At 16K, Full is also more robust than SWA, while GDN remains close to Full. Within hybrids, GDN achieves higher sparse scores than SWA in most evaluated settings.

Full's stronger sparse robustness, especially at longer contexts, motivates addressing the entropy bias of hybrid full-attention layers during training.

4 Concentration Prior for Sparse-Ready Attention

4.1 Retained Mass and Sparse Approximation

Let $\boldsymbol{a}$ be one dense attention row and $\boldsymbol{a}^{(k)}$ its renormalized oracle top-k approximation. For the same value vectors, write $\boldsymbol{o} = \sum_i a_i \boldsymbol{v}_i$ and $\boldsymbol{o}^{(k)} = \sum_i a_i^{(k)} \boldsymbol{v}_i$. If $\|\boldsymbol{v}_i\|_2 \le B$, then

$$\begin{aligned} \|\boldsymbol{a} - \boldsymbol{a}^{(k)}\|_1 &= 2\big(1 - M_k(\boldsymbol{a})\big), \\ \|\boldsymbol{o} - \boldsymbol{o}^{(k)}\|_2 &\le 2B\big(1 - M_k(\boldsymbol{a})\big). \end{aligned} \tag{9}$$

The first identity accounts for both the discarded mass and the renormalization of selected keys; the second follows by the triangle inequality. Thus, higher retained mass reduces the change in attention weights and tightens the output-error bound at a fixed sparse budget.

4.2 From Logit Scale to Concentration

Consider fixed logits $\boldsymbol{z}$ and $\boldsymbol{a}(g) = \mathrm{softmax}(g\boldsymbol{z})$ for $g > 0$. Writing $Z(g) = \sum_i \exp(gz_i)$ and $\mu(g) = \sum_i a_i(g) z_i$, we have $a_i'(g) = a_i(g)(z_i - \mu)$ and $\mathcal{H}(\boldsymbol{a}(g)) = \log Z(g) - g\mu(g)$. Differentiating gives

$$\frac{\partial \mathcal{H}(\boldsymbol{a}(g))}{\partial g} = -g \, \mathrm{Var}_{\boldsymbol{a}(g)}[z] \le 0. \tag{10}$$

Positive scaling also preserves the top-k set. For $1 \le k < n$, let μ_{in} and μ_{out} be the attention-weighted mean logits within and outside that set. Its retained mass satisfies

$$\frac{\partial M_k(g)}{\partial g} = M_k(g)\big(1 - M_k(g)\big)(\mu_{\mathrm{in}} - \mu_{\mathrm{out}}) \ge 0, \tag{11}$$

because selected logits are no smaller than excluded logits. Therefore, increasing the scale simultaneously lowers entropy and raises top-k retained mass, bringing the scaled dense distribution closer to its own sparse approximation. This provides a concrete motivation for a concentration prior: encourage attention distributions whose outputs are less perturbed by removing low-ranked keys. Complete proofs are provided in Appendix E.

4.3 Applying the Prior During Pretraining

QKNorm separates normalized query–key similarity from logit scale (Equation 7). Our implementation uses queries $\hat{\boldsymbol{q}}$ and keys $\hat{\boldsymbol{k}}$ after positional encoding and RMS normalization without learned gains. For hybrid full-attention layers $\mathcal{F}$, we compute

$$a_{\ell,t,s} = \mathrm{softmax}_{s \le t}\left(g \, \frac{\hat{\boldsymbol{q}}_{\ell,t}^{\top} \hat{\boldsymbol{k}}_{\ell,s}}{\sqrt{d_k}} \right), \qquad \ell \in \mathcal{F}. \tag{12}$$

Here $g > 0$ is the product scale, equivalent to multiplying normalized Q and K by $\sqrt{g}$ each. It is shared across full-attention layers and fixed during pretraining and evaluation; efficient-attention components are unchanged.

We fix g to control differences in independently learned scales, using the same value in matched architectural comparisons and varying it in scale ablations. Learned-gain alternatives are examined in Section 5.4. Training retains the standard language-modeling objective without an auxiliary entropy loss, allowing the QK representations to adapt to the prior throughout pretraining. Unlike inference-only sharpening, this shapes the learning process under the same concentration setting used for evaluation, without changing the trained attention distribution afterward. Its effect on learned attention distributions and sparse performance is evaluated empirically.

5 Experiments

5.1 Experimental Setup

Architecture. We compare three decoder-only architectures, denoted *Full*, *SWA*, and *GDN* throughout the paper. Full uses full attention in every layer. The two hybrid architectures use a fixed 3:1 ratio of efficient-attention to full-attention layers. SWA uses sliding-window attention with a window size of 512, whereas GDN uses Gated DeltaNet with 16 key heads, 16 value heads, and a width-4 depthwise causal convolution. All softmax-attention layers use grouped-query attention (GQA).

Pretraining. Our pretrained models span four sizes, S1–S4, whose configurations are given in Table 1; unless otherwise specified, experiments use the largest S4 model. We pretrain with a 16K context for $D = 100N$ tokens, where N is the model parameter count excluding token and value embeddings; GDN uses the same explicit token budget as the matched Full and SWA models. For the standard 64K extension, we continue pretraining for an additional 34.9B tokens and increase the RoPE base from 10^5 to 8×10^5. For the indexer-based experiments, we subsequently perform two-stage sparse training following DeepSeek Sparse Attention (DSA) (DeepSeek-AI, 2025), with a top-k budget of 2,048. We use 0.5B tokens for indexer warmup, followed by a 10B-token sparse continued-pretraining budget to adapt the model to sparse attention. For the controlled pretraining comparisons, all architectures and scale variants use the same corpus, example order, token budget, and optimization schedule.

Size	Layers	Hidden size	Q/KV heads	Head dim.	Total params	Params (w/o embed.)	Training tokens
S1	12	768	12/2	64	148.6M	98.3M	9.8B
S2	16	1,152	12/2	96	345.3M	257.2M	25.7B
S3	20	1,536	16/4	96	722.0M	545.8M	54.6B
S4	24	1,536	16/4	96	846.2M	644.9M	64.5B

Table 1: Key hyperparameters of the scaling models.

Evaluation. We primarily use RULER (Hsieh et al., 2024) to evaluate long-context ability, with LongBench (Bai et al., 2024) as a complementary long-context benchmark. Except in Section 5.5, top-k denotes oracle per-head selection of the k keys with the highest dense attention scores in each sparsified full-attention layer. General ability is measured by an aggregate score over 22 tasks, including HellaSwag, ARC-Challenge, CommonsenseQA, and SQuAD.

Further details of the architectures, pretraining, and evaluation are provided in Appendix A.

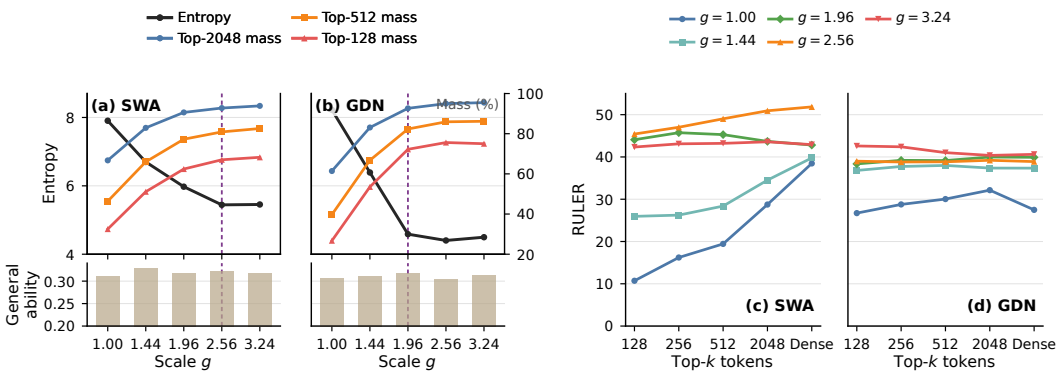

Figure 4: Effect of the concentration prior on (a–b) entropy, retained mass, and general ability, and (c–d) dense and sparse 16K RULER scores. Dashed lines mark the approximate onset of entropy saturation.

5.2 Effect of the Concentration Prior

We evaluate five full-attention logit scales, $g \in \{1.00, 1.44, 1.96, 2.56, 3.24\}$ for both hybrid architectures while holding the remaining training configuration fixed. Figure 4 reports attention concentration, general ability, and dense and sparse 16K RULER performance; per-task scores are provided in Appendix B.

The prior directly controls attention concentration. Increasing g lowers entropy and raises retained mass. For SWA, entropy decreases from 7.90 at $g = 1.00$ to 5.44 at $g = 2.56$, while top-128 mass increases from 32.5% to 67.0%. For GDN, entropy decreases from 8.22 to 4.59 at $g = 1.96$, with top-128 mass increasing from 26.7% to 72.2%.

Concentrated attention substantially improves sparse long-context performance. At $g = 1.00$, top-128 reduces SWA's RULER score from 38.5 to 10.7. At $g = 1.96$, the sparse score reaches 44.1 versus 42.8 with dense attention; $g = 2.56$ gives the strongest SWA scores, 51.8 dense and 45.4 at top-128. GDN's top-128 score also improves, from 26.7 at $g = 1.00$ to 38.4 at $g = 1.96$. Increased retained mass thus accompanies substantially better long-context performance under sparse attention.

Entropy saturation guides the choice of scale. Entropy initially falls sharply but approaches a plateau: SWA changes from 5.44 at $g = 2.56$ to 5.45 at $g = 3.24$, while GDN stays near 4.4–4.6 beyond $g = 1.96$. Long-context performance no longer improves consistently in this regime. During the initial entropy reduction, general ability instead improves slightly, from 0.311 to 0.328 for SWA and from 0.306 to 0.318 for GDN. Thus, better retrieval and general ability need not be in opposition. Beyond saturation, further scaling can degrade general ability: GDN falls from 0.318 at $g = 1.96$ to 0.303 at $g = 2.56$, with a similar decline in SWA. We conjecture that an overly strong prior may interfere with learning once further sharpening offers little benefit. Entropy saturation therefore provides a practical guide to choosing the scale. In practice, the useful range is identified by the joint response of entropy, sparse long-context performance, and general ability, rather than by minimizing entropy alone. The different saturation points of SWA and GDN also suggest that prior strength should be calibrated to the architecture.

5.3 Effective Application of the Concentration Prior

We examine where to apply the prior, whether its benefit persists across model sizes, and when it should be introduced during pretraining. All ablations in Table 2 use SWA hybrids; the main comparison is the RULER score retained under sparse full attention.

Apply the prior to full-attention layers. At both tested scales, applying g only to full-attention layers yields a higher top-128 RULER score than applying it to every attention layer: 45.42 versus

Configuration	RULER@Dense	RULER@2048	RULER@512	RULER@128	GA
Application scope					
$g = 2.56$ @ all	36.86	37.69	37.40	35.86	0.321
$g = 2.56$ @ full	**51.84**	**50.92**	**49.02**	**45.42**	**0.323**
$g = 3.24$ @ all	39.18	39.37	39.29	39.13	**0.340**
$g = 3.24$ @ full	**42.92**	**43.58**	**43.22**	**42.34**	0.318
Model size					
S1 @ $g = 1.00$	14.30	14.36	13.27	8.47	0.159
S1 @ $g = 2.56$	**18.03**	**18.65**	**19.41**	**19.64**	**0.179**
S2 @ $g = 1.00$	24.28	20.88	16.29	8.90	0.238
S2 @ $g = 2.56$	**37.88**	**35.99**	**35.78**	**35.27**	**0.249**
S3 @ $g = 1.00$	36.54	33.30	30.15	20.85	0.310
S3 @ $g = 2.56$	**41.94**	**42.04**	**42.26**	**41.49**	**0.319**
Prior introduction during pretraining					
$1.44 \to 2.56$ @ 0/4	**51.84**	**50.92**	**49.02**	**45.42**	0.323
$1.44 \to 2.56$ @ 1/4	41.69	42.21	43.31	40.35	**0.335**
$1.44 \to 2.56$ @ 2/4	45.42	44.20	42.12	39.72	0.328
$1.44 \to 2.56$ @ 3/4	44.65	43.11	38.77	35.21	**0.335**
$1.44 \to 2.56$ @ 4/4	39.77	34.52	28.38	25.95	0.328

Table 2: Concentration-prior ablations at 16K: application scope, model size, and introduction time. RULER@k denotes the score with top-k attention; GA denotes general ability. $x/4$ indicates the fraction of pretraining completed before switching from $g = 1.44$ to $g = 2.56$.

35.86 at $g = 2.56$, and 42.34 versus 39.13 at $g = 3.24$. This advantage does not extend uniformly to general ability, which is higher for the all-layer variant at $g = 3.24$. These results favor applying the concentration prior only to full-attention layers for sparse long-context performance.

The benefit persists across model sizes. For S1–S3, applying $g = 2.56$ raises top-128 RULER by 11.17, 26.37, and 20.64 points, respectively, relative to $g = 1.00$. The corresponding dense gains are smaller at 3.73, 13.60, and 5.40 points, while general ability improves slightly at every size. Thus, the concentration prior remains useful across the evaluated model sizes, particularly under a tight sparse budget.

Earlier introduction yields stronger sparse performance. When the change from $g = 1.44$ to $g = 2.56$ is delayed, top-128 RULER declines from 45.42 with $g = 2.56$ throughout pretraining to 40.35, 39.72, and 35.21 after introduction at approximately one-, two-, and three-quarters of pretraining; remaining at $g = 1.44$ gives 25.95. The three switched variants share the same final scale, indicating that sparse capability depends on how long the model is trained with the prior, not only on its final logit scale. Even the latest switch improves top-128 performance over never switching, whereas general ability shows no monotonic trend with introduction time.

5.4 Long-Context Extension and Learned QK Scaling

We next ask whether the prior remains effective after extending the context to 64K, and whether learned QK normalization gains can substitute for it. Table 3 reports two separate SWA-hybrid comparisons on 64K RULER.

The benefit persists at 64K. With a fixed full-layer scale, increasing g from 1.00 to 2.56 raises dense RULER from 19.22 to 36.65 and top-128 RULER from 2.12 to 32.76. The drop from dense to top-128 contracts from 17.10 to 3.89 points, and gains are also present at top-2048 and top-512. Thus, the concentration prior improves both long-context performance and its preservation under sparse attention after the 64K extension.

Configuration	RULER@Dense	RULER@2048	RULER@512	RULER@128
Fixed full-layer scale				
$g = 1.00$	19.22	9.87	3.30	2.12
$g = 2.56$	**36.65**	**36.24**	**35.80**	**32.76**
Learned QK scaling				
Learn-QK	31.13	29.43	25.49	23.27
Learn-QK + $g = 1.96$	**33.64**	**33.81**	**32.61**	**31.58**

Table 3: 64K RULER scores with fixed and learned QK scaling in SWA hybrids. Dense denotes native attention; RULER@k denotes top-k attention.

Benchmark	Fixed $g = 1.44$	Fixed $g = 2.56$	Learn-QK	Learn-QK + $g = 1.96$
RULER	26.57	**35.67**	28.13	**32.92**
LongBench	21.74	**21.81**	20.54	**20.91**

Table 4: Mean 64K RULER and LongBench scores with trained top-2,048 indexers.

Learned QK scaling does not remove the benefit. We also test an SWA hybrid with learned per-channel gains in QK RMS normalization. Both learned-QK variants use the same language-modeling objective, which does not explicitly reward retaining attention mass at a fixed sparse budget. Adding an explicit full-layer product scale of $g = 1.96$ during 64K continued pretraining raises its dense score from 31.13 to 33.64 and its top-128 score from 23.27 to 31.58. The corresponding dense-to-top-128 drop shrinks from 7.86 to 2.06 points. Independently, a matched-layer analysis at a later 64K checkpoint finds that learned QK gains leave the hybrid's full-attention layers with higher entropy and lower top-k mass than their Full counterparts (Appendix D). Thus, learned QK gains do not fully substitute for an explicit concentration prior in these experiments.

5.5 Evaluation with a Trained Sparse Indexer

We extend the oracle top-k experiments to trained indexer-based sparsity, evaluating four 64K SWA hybrids after sparse continued pretraining. Each full-attention layer uses a shared indexer to select 2,048 keys across heads. We compare fixed scales of $g = 1.44$ and $g = 2.56$, and learned QK gains with and without an additional concentration prior of $g = 1.96$, on RULER and LongBench. Detailed results are provided in Appendix G.

Table 4 shows that increasing the fixed scale raises RULER from 26.57 to 35.67; adding the prior to learned QK gains raises it from 28.13 to 32.92. The gains are strongest on retrieval tasks, with mean NIAH scores increasing from 36.56 to 50.31 and from 37.94 to 46.31, respectively. LongBench averages improve only slightly, with mixed changes across task categories. Together with the oracle ablations, these results support the importance of attention concentration for sparse long-context retrieval when selection is performed by a learned indexer.

6 Conclusion

We identify a systematic entropy bias in hybrid architectures: their full-attention layers develop more diffuse attention and are less robust to sparsification than those of full-attention models. Our experiments show that a fixed concentration prior during pretraining mitigates this bias, increases retained attention mass, and improves sparse long-context retrieval. The benefit persists across model sizes and context extension, and remains useful with learned QK gains and trained sparse indexers. Entropy saturation provides a practical guide to balancing sparse performance and general ability. These findings highlight attention concentration as an architecture-aware training consideration for sparse-ready hybrid models.

AI USE STATEMENT

Generative AI tools assisted with drafting selected passages, polishing the writing, and preparing experimental code. The human authors developed the core idea, designed the experiments, established the paper's overall structure and argument, and checked the correctness of the code and results. The human authors reviewed the final manuscript and take responsibility for its claims.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 3119–3137, 2024.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

David Chiang and Peter Cholak. Overcoming a theoretical limitation of self-attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, pp. 10041–10071. PMLR, 2024.

DeepSeek-AI. DeepSeek-V3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, et al. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. *Online technical report*, 2024.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, et al. MoBA: Mixture of block attention for long-context LLMs. *arXiv preprint arXiv:2502.13189*, 2025.

Ken M. Nakanishi. Scalable-softmax is superior for attention. In *International Conference on Machine Learning*, 2025.

Ziqing Qiao, Yinuo Xu, Chaojun Xiao, Zhou Su, Zihan Zhou, Yingfa Chen, Xiaoyue Xu, Xu Han, and Zhiyuan Liu. Rethinking the role of efficient attention in hybrid architectures. *arXiv preprint arXiv:2606.15378*, 2026.

Zhen Qin, Weigao Sun, Dong Li, Xuyang Shen, Weixuan Sun, and Yiran Zhong. Various lengths, constant speed: Efficient language modeling with lightning attention. In *International Conference on Machine Learning*, pp. 41517–41535. PMLR, 2024.

Qwen Team. Qwen3.5. Qwen Blog, 2026a. URL `https://qwen.ai/blog?id=qwen3.5`.

Qwen Team. Qwen3.8-Flash-Next: A new architecture, towards ultimate cost-efficiency. Qwen Blog, 2026b. URL `https://qwen.ai/blog?id=qwen3.8-flash-next`.

Jianlin Su. Understanding attention scaling through entropy invariance. Scientific Spaces blog, 2021. URL https://kexue.fm/archives/8823.

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.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, et al. MiMo-V2-Flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, 2025.

Z.ai. GLM-5.3-Flash: Frontier intelligence, flash cost. Z.ai Blog, 2026. URL https://z.ai/blog/glm-5.3-flash.

Wen Zan, Jiaqi Zhang, Jianchao Tan, Hong Liu, Cunguang Wang, et al. LongCat Sparse Attention: Taming the lightning via streaming-aware hierarchical cross-layer indexing. *arXiv preprint arXiv:2608.01662*, 2026.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Josh Susskind. Stabilizing transformer training by preventing attention entropy collapse. *arXiv preprint arXiv:2303.06296*, 2023.

Weilin Zhao, Zihan Zhou, Zhou Su, Chaojun Xiao, Yuxuan Li, et al. InfLLM-V2: Dense-sparse switchable attention for seamless short-to-long adaptation. *arXiv preprint arXiv:2509.24663*, 2025.

A Training and Architecture Details

A.1 Architecture Configurations

The Full baseline places full attention in every layer. The SWA hybrid follows an SSSL pattern: three sliding-window attention layers with window size 512 are followed by one full-attention layer. The GDN hybrid follows the analogous GGGL pattern, replacing each SWA layer with a Gated DeltaNet layer while retaining the same full-attention locations. Table 5 summarizes the S4 architectures.

Model	Pattern	Full layers	Efficient component
Full	L	24	–
SWA hybrid	SSSL	6	SWA ($w = 512$)
GDN hybrid	GGGL	6	Gated DeltaNet

Table 5: S4 attention patterns: L, S, and G denote full attention, sliding-window attention, and Gated DeltaNet, respectively.

The Gated DeltaNet layers use 16 key heads and 16 value heads with head dimension 128 and value expansion factor 1. Each layer includes a width-4 depthwise causal convolution and an output gate. The surrounding Transformer backbone and placement of full-attention layers are shared with the other S4 models.

A.2 Data and Training Schedule

The 16K pretraining corpus mixes ClimbMix with long-document chunks from ProLong at a 70/30 token ratio. Sources are mixed at the example level before packing. Table 6 gives the sequence length, global token batch, and total token budget for every scale. Within each matched comparison, runs consume the same shuffled example order.

For the standard 64K stage, we continue from the completed S4 16K checkpoints using shuffled and repacked ProLong documents and increase the RoPE base from 10^5 to 8×10^5. Model parameters and the concentration scale assigned during 16K pretraining are otherwise preserved.

A.3 Optimization

We use a combined Muon–AdamW optimizer (Jordan et al., 2024). Two-dimensional Transformer matrix parameters use Muon with base learning rate 0.02 and weight decay 0.28; embedding, output, and scalar parameters use AdamW with base learning rates 0.3, 0.008, and 0.5, respectively. The embedding and output learning rates are scaled by $(d_{\text{model}}/768)^{-1/2}$. We use a Warmup–Stable–Decay schedule (Hu et al., 2024): the learning rate warms up linearly for 40 updates, remains constant, and decays linearly over the final 65% of training to 5% of its peak value. Muon momentum increases from 0.85 to 0.97 over the first 400 updates and decays to 0.90 during the learning-rate decay phase.

Parameters inside Gated DeltaNet blocks use AdamW with base learning rate 3×10^{-4} (effective learning rate 6×10^{-4} after the existing optimizer scaling), $\beta = (0.9, 0.95)$, $\epsilon = 10^{-15}$, and weight decay 0.1 for matrix and convolution weights. All remaining GDN-hybrid parameters use the same optimizer groups as the Full and SWA models.

A.4 Sparse Continued Pretraining

Indexer architecture. We adapt the two-stage training recipe of DSA (DeepSeek-AI, 2025) to the 64K SWA hybrid checkpoints. Each of the six full-attention layers has an independent indexer with four query heads, one shared key head, and head dimension 32. Bias-free linear projections produce the indexer queries, keys, and query-dependent head weights from the layer's hidden states. Queries and keys are normalized with LayerNorm; the indexer does not use RoPE. Its score for query t and key s is

$$I_{t,s} = \sum_{r=1}^{4} w_{t,r}^{I} \, \mathrm{LeakyReLU}_{0.1}\left(\frac{(q_{t,r}^{I})^{\top} k_{s}^{I}}{\sqrt{32}} \right), \tag{13}$$

Stage	Seq. length	Token batch	Tokens	RoPE base	Initialization
S1 16K	16,384	1,048,576	9.8B	10^5	Scratch
S2 16K	16,384	2,097,152	25.7B	10^5	Scratch
S3 16K	16,384	2,097,152	54.6B	10^5	Scratch
S4 16K	16,384	2,097,152	64.5B	10^5	Scratch
S4 64K extension	65,536	2,097,152	34.9B	8×10^5	S4 16K

Table 6: Training budgets for the scaling study and the 64K extension. The token count in the last row is additional to the S4 16K pretraining budget.

where the head weights are unconstrained linear outputs and LeakyReLU uses negative slope 0.1. After causal masking, the indexer selects up to 2,048 keys per query; this selection is shared by all 16 main attention heads in that layer. The 18 sliding-window layers retain their window size of 512.

Stage 1: indexer warmup. We freeze the pretrained backbone and retain dense attention while training the indexers for 0.5B tokens. At each query, the teacher distribution is the average of the 16 main heads' dense causal attention distributions. The student distribution is the softmax of the indexer scores over all causally visible keys. We minimize the KL divergence from the detached teacher to the student, averaged over query positions and full-attention layers. Only the indexer parameters are updated.

Stage 2: sparse adaptation. We initialize the indexers from Stage 1, unfreeze the backbone, and continue training with a 10B-token budget. The full-attention layers now use the indexers' actual top-2,048 selections for the autoregressive language-modeling objective. The indexers continue to minimize a KL objective restricted to the selected keys: each main attention head is normalized over this selected support before its distribution is averaged to form the teacher, and the indexer scores are normalized over the same support. The discrete selections and teacher distributions are detached. We also detach the hidden-state inputs to the indexers, so the language-modeling loss updates the backbone and the KL loss updates the indexers without backpropagating into the backbone. Both stages use 64K sequences from the shuffled and repacked ProLong data.

Sparse-training optimization. Indexer warmup uses AdamW with learning rate 3×10^{-4}, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and zero weight decay. During sparse adaptation, the backbone uses AdamW with learning rate 10^{-5}, $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, and weight decay 0.1; the indexers use a separate AdamW optimizer with learning rate 10^{-4} and the same other settings as warmup. Gradients are clipped to norm 1.0 for each optimizer.

A.5 Evaluation Protocol

For RULER, we evaluate 13 tasks at 16K or 64K context length using 50 examples per task and the same fixed inputs across matched models at each length. We report the equally weighted mean over the 13 task scores. Unless otherwise specified, sparse runs use exact per-head top-k selection in full-attention layers. Section 5.5 uses the trained shared indexers described in Appendix A.4.

LongBench v1 contains 21 tasks (4,750 examples per model). We use the official task prompts, per-task generation limits, and metrics with greedy decoding, and truncate overlength inputs symmetrically from the middle to fit the 64K context. All evaluated checkpoints are base language models without instruction tuning. The overall LongBench score is the equally weighted mean over its 21 tasks.

For general ability, we evaluate 22 short-context tasks, with at most 500 examples per task, and report their equally weighted mean after centering each task against its random baseline. If a is task accuracy and r is the random baseline as a fraction, the centered score is $(a - r)/(1 - r)$. The tasks are grouped into language understanding (6), world knowledge (4), commonsense reasoning (4), symbolic problem solving (5), and reading comprehension (3). The complete task list and per-task scores are in Table 7.

B Scale-Ablation Benchmark Breakdowns

B.1 General-Ability Breakdown

Table 7 reports centered scores for all 22 general-ability tasks, grouped by category, and the aggregate used in Figure 4.

	SWA hybrid					GDN hybrid				
Task	1.00	1.44	1.96	2.56	3.24	1.00	1.44	1.96	2.56	3.24
Language understanding (6)										
HellaSwag (zero-shot)	0.507	0.515	0.472	0.483	0.493	0.496	0.483	0.496	0.493	0.499
LAMBADA	0.596	0.618	0.584	0.618	0.580	0.582	0.566	0.598	0.592	0.618
HellaSwag	0.509	0.523	0.509	0.539	0.525	0.504	0.509	0.493	0.507	0.504
Winograd	0.524	0.473	0.473	0.516	0.502	0.487	0.502	0.531	0.465	0.473
WinoGrande	0.180	0.160	0.168	0.156	0.192	0.188	0.184	0.208	0.132	0.148
Language ID	0.175	0.186	0.182	0.138	0.164	0.140	0.162	0.182	0.171	0.162
World knowledge (4)										
Jeopardy	0.222	0.244	0.234	0.240	0.224	0.214	0.220	0.262	0.230	0.206
Wikidata QA	0.552	0.558	0.530	0.560	0.550	0.544	0.538	0.544	0.540	0.576
ARC-Easy	0.648	0.664	0.667	0.643	0.656	0.603	0.637	0.645	0.624	0.645
ARC-Challenge	0.232	0.269	0.283	0.269	0.259	0.237	0.248	0.269	0.253	0.272
Commonsense reasoning (4)										
COPA	0.380	0.300	0.380	0.420	0.380	0.420	0.440	0.360	0.400	0.360
CommonsenseQA	0.037	0.163	0.087	0.058	0.020	0.018	0.140	0.045	0.085	0.100
PIQA	0.548	0.556	0.540	0.556	0.584	0.544	0.528	0.564	0.564	0.536
OpenBookQA	0.235	0.216	0.197	0.235	0.237	0.240	0.213	0.240	0.227	0.227
Symbolic problem solving (5)										
Dyck languages	0.106	0.076	0.092	0.112	0.080	0.058	0.062	0.068	0.058	0.096
LSAT-AR	0.060	0.054	0.038	0.098	0.076	0.049	0.027	0.103	0.071	0.087
CS algorithms	0.424	0.474	0.426	0.466	0.480	0.402	0.436	0.470	0.448	0.470
Operators	0.219	0.200	0.229	0.186	0.210	0.181	0.195	0.171	0.195	0.210
Repeat-copy logic	0.062	0.062	0.094	0.031	0.062	0.000	0.031	0.000	0.031	0.031
Reading comprehension (3)										
SQuAD	0.536	0.540	0.540	0.510	0.552	0.494	0.526	0.540	0.536	0.544
CoQA	0.326	0.324	0.340	0.334	0.346	0.320	0.300	0.326	0.322	0.310
BoolQ	-0.237	0.042	-0.105	-0.068	-0.179	0.011	-0.100	-0.121	-0.274	-0.179
Overall	0.311	0.328	0.316	0.323	0.318	0.306	0.311	0.318	0.303	0.313

Table 7: Centered general-ability scores across concentration scales.

B.2 RULER Breakdown

Tables 8 and 9 report all 13 task scores underlying the aggregate curves in Figure 4. Avg. averages all 13 tasks; NIAH avg. and non-NIAH avg. average the eight needle-retrieval and five remaining tasks, respectively. Task abbreviations are S1–S3 (single-key retrieval), MK1–MK3 (multi-key retrieval), MV (multi-value retrieval), MQ (multi-query retrieval), VT (variable tracking), CWE/FWE (common/frequent-word extraction), and QA1/QA2 (question answering).

g	Attention	Avg.	NIAH avg.	Non-NIAH avg.	S1	S2	S3	MK1	MK2	MK3	MV	MQ	VT	CWE	FWE	QA1	QA2
1.00	Dense	38.47	49.69	20.52	98.0	80.0	82.0	74.0	2.0	2.0	28.5	31.0	20.0	2.6	36.0	18.0	26.0
	2048	28.75	37.19	15.24	100.0	40.0	56.0	48.0	0.0	0.0	25.5	28.0	2.8	1.4	38.0	14.0	20.0
	512	19.43	24.31	11.63	92.0	24.0	22.0	22.0	0.0	0.0	19.0	15.5	4.4	2.4	17.3	14.0	20.0
	256	16.21	19.94	10.25	82.0	22.0	10.0	16.0	0.0	0.0	17.0	12.5	10.8	3.8	4.7	10.0	22.0
	128	10.71	13.00	7.05	58.0	18.0	2.0	6.0	0.0	0.0	12.0	8.0	14.8	3.8	6.7	4.0	6.0
1.44	Dense	39.77	57.25	11.80	100.0	100.0	86.0	78.0	22.0	2.0	36.5	33.5	2.4	2.6	0.0	22.0	32.0
	2048	34.52	49.06	11.24	100.0	72.0	76.0	70.0	14.0	0.0	31.5	29.0	1.2	5.0	0.0	20.0	30.0
	512	28.38	39.81	10.08	100.0	58.0	42.0	60.0	4.0	0.0	30.0	24.5	2.0	4.4	0.0	18.0	26.0
	256	26.24	36.75	9.43	100.0	54.0	32.0	58.0	4.0	0.0	26.5	19.5	4.0	5.8	1.3	16.0	20.0
	128	25.95	36.25	9.47	98.0	56.0	38.0	52.0	4.0	0.0	23.5	18.5	4.4	5.6	1.3	16.0	20.0
1.96	Dense	42.81	55.31	22.81	100.0	96.0	82.0	78.0	16.0	0.0	30.5	40.0	9.2	2.2	46.7	32.0	24.0
	2048	43.68	56.06	23.88	100.0	96.0	80.0	78.0	26.0	0.0	29.0	39.5	13.6	1.8	46.0	32.0	26.0
	512	45.28	58.50	24.13	100.0	96.0	82.0	80.0	36.0	2.0	33.0	39.0	16.0	4.0	44.7	34.0	22.0
	256	45.74	60.00	22.92	100.0	98.0	86.0	76.0	40.0	2.0	34.0	44.0	22.0	4.6	36.0	34.0	18.0
	128	44.08	58.75	20.60	100.0	96.0	88.0	70.0	32.0	2.0	38.5	43.5	25.2	3.8	24.0	30.0	20.0
2.56	Dense	51.84	68.19	25.69	100.0	98.0	80.0	78.0	54.0	48.0	44.5	43.0	26.4	1.4	40.7	32.0	28.0
	2048	50.92	68.56	22.69	100.0	96.0	80.0	80.0	56.0	50.0	45.5	41.0	25.2	1.6	30.7	30.0	26.0
	512	49.02	68.44	17.96	100.0	92.0	90.0	78.0	54.0	46.0	46.0	41.5	23.6	2.2	12.0	26.0	26.0
	256	47.03	66.50	15.87	100.0	94.0	94.0	80.0	40.0	38.0	44.0	42.0	26.0	4.0	3.3	24.0	22.0
	128	45.42	65.06	14.00	98.0	96.0	94.0	74.0	34.0	36.0	44.0	44.5	26.4	5.6	0.0	24.0	14.0
3.24	Dense	42.92	58.94	17.29	100.0	94.0	82.0	76.0	26.0	16.0	40.0	37.5	30.8	3.0	0.7	22.0	30.0
	2048	43.58	59.94	17.41	100.0	94.0	82.0	72.0	28.0	18.0	44.5	41.0	32.0	2.4	0.7	20.0	32.0
	512	43.22	59.13	17.76	100.0	96.0	84.0	68.0	24.0	16.0	42.0	43.0	34.0	2.8	2.0	20.0	30.0
	256	43.10	58.50	18.47	100.0	96.0	82.0	74.0	22.0	10.0	40.5	43.5	36.0	3.0	3.3	20.0	30.0
	128	42.34	57.88	17.49	100.0	100.0	82.0	70.0	16.0	8.0	42.5	44.5	30.4	2.4	4.7	20.0	30.0

Table 8: Per-task 16K RULER scores for SWA across scales and attention budgets.

g	Attention	Avg.	NIAH avg.	Non-NIAH avg.	S1	S2	S3	MK1	MK2	MK3	MV	MQ	VT	CWE	FWE	QA1	QA2
1.00	Dense	27.48	37.88	10.85	48.0	80.0	42.0	74.0	2.0	0.0	29.5	27.5	0.0	1.6	0.7	24.0	28.0
	2048	32.15	44.31	12.68	56.0	90.0	66.0	78.0	2.0	0.0	34.5	28.0	0.0	1.4	0.0	34.0	28.0
	512	30.04	42.94	9.40	46.0	84.0	68.0	74.0	0.0	0.0	39.0	32.5	0.0	1.0	0.0	22.0	24.0
	256	28.77	40.88	9.40	46.0	78.0	58.0	72.0	0.0	0.0	37.5	35.5	0.0	1.0	0.0	22.0	24.0
	128	26.72	39.81	5.76	48.0	72.0	62.0	68.0	0.0	0.0	34.5	34.0	0.0	0.8	0.0	18.0	10.0
1.44	Dense	37.35	52.50	13.12	84.0	96.0	82.0	74.0	4.0	6.0	37.5	36.5	0.8	4.8	2.0	32.0	26.0
	2048	37.37	52.50	13.16	82.0	96.0	80.0	76.0	4.0	8.0	38.0	36.0	0.0	3.8	4.0	34.0	24.0
	512	37.99	52.75	14.39	78.0	96.0	80.0	78.0	2.0	8.0	41.5	38.5	2.8	5.8	1.3	38.0	24.0
	256	37.75	53.06	13.24	76.0	96.0	80.0	80.0	2.0	8.0	44.0	38.5	0.4	5.8	0.0	36.0	24.0
	128	36.81	53.44	10.20	76.0	96.0	82.0	82.0	0.0	8.0	46.5	37.0	0.0	7.0	0.0	30.0	14.0
1.96	Dense	39.91	53.63	17.97	82.0	92.0	78.0	84.0	10.0	4.0	43.5	35.5	1.2	2.0	26.7	34.0	26.0
	2048	39.98	53.81	17.85	82.0	96.0	78.0	84.0	10.0	4.0	41.5	35.0	2.4	2.2	24.7	34.0	26.0
	512	39.15	55.06	13.69	82.0	96.0	84.0	84.0	14.0	4.0	40.5	36.0	2.4	5.4	8.7	30.0	22.0
	256	39.20	55.13	13.72	82.0	94.0	86.0	86.0	14.0	4.0	39.5	35.5	4.4	6.2	4.0	32.0	22.0
	128	38.35	54.13	13.12	82.0	90.0	88.0	84.0	14.0	4.0	37.0	34.0	3.2	6.4	4.0	34.0	18.0
2.56	Dense	38.90	52.81	16.64	96.0	100.0	80.0	74.0	4.0	2.0	31.5	35.0	10.0	1.2	4.0	44.0	24.0
	2048	39.21	53.13	16.95	96.0	100.0	80.0	78.0	4.0	2.0	31.0	34.0	8.0	1.4	3.3	44.0	28.0
	512	38.85	53.50	15.41	96.0	100.0	84.0	78.0	4.0	2.0	30.0	34.0	6.8	1.6	2.7	40.0	26.0
	256	38.81	54.25	14.11	96.0	100.0	86.0	80.0	4.0	4.0	28.5	35.5	5.2	2.0	3.3	36.0	24.0
	128	39.00	55.19	13.09	98.0	100.0	86.0	76.0	6.0	10.0	30.0	35.5	6.8	2.0	0.7	32.0	24.0
3.24	Dense	40.62	55.63	16.60	88.0	100.0	88.0	68.0	20.0	0.0	39.5	41.5	13.2	1.8	0.0	34.0	34.0
	2048	40.37	55.94	15.45	88.0	100.0	88.0	70.0	20.0	0.0	37.5	44.0	15.2	1.4	0.7	30.0	30.0
	512	41.02	56.50	16.25	86.0	100.0	86.0	78.0	20.0	0.0	39.0	43.0	15.6	1.0	0.7	32.0	32.0
	256	42.40	57.25	18.64	86.0	100.0	88.0	82.0	20.0	0.0	41.5	40.5	24.4	0.8	0.0	36.0	32.0
	128	42.60	57.69	18.45	86.0	100.0	90.0	84.0	18.0	4.0	40.0	39.5	26.4	1.2	0.7	32.0	32.0

Table 9: Per-task 16K RULER scores for GDN across scales and attention budgets.

C Additional Retrieval-Head Analysis

The retrieval probe contains 50 fixed examples from each of eight 16K RULER needle-in-a-haystack tasks. We append the answer prefix and measure, at the final prefill query, the attention mass assigned to all gold-answer tokens. We also evaluate a length-matched, non-overlapping control span at another location in the haystack; its mean mass is near 5×10^{-4} for all three models.

Figure 5 extends the matched-layer maps in Figure 2(b) to all 24 layers of the Full model. In Full, retrieval-active heads are spread across several middle and upper layers. The SWA hybrid concentrates its strongest clusters at layers 15 and 19, while the GDN hybrid shifts its strongest cluster to layer 11 and retains weaker retrieval signals across later full-attention layers. Gray rows mark efficient-attention layers. All panels share a logarithmic color scale; outlined cells have mean answer mass above 0.02.

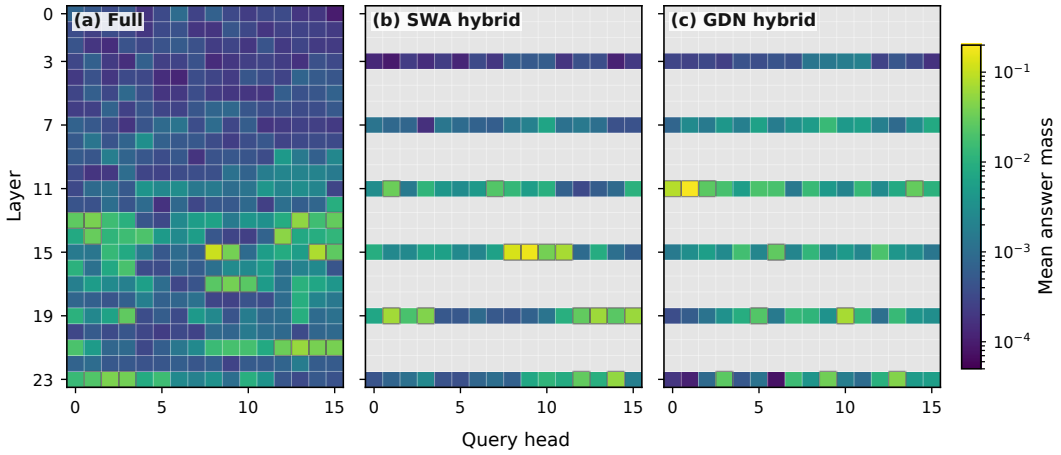

Figure 5: Complete layer–head maps of mean answer-token attention mass.

D Learned QK Scaling and Attention Concentration at 64K

Our Learn-QK variant uses RMS normalization with trainable per-channel Q/K gains; the additional concentration scale g acts on top of these gains.

Both the fixed-prior and Learn-QK variants use the language-modeling objective during pretraining and 64K continuation. Learned gains are optimized by this loss, whereas g is explicitly controlled to study concentration and sparse performance. We examine whether the learned gains eliminate the hybrid's entropy bias and whether an additional concentration prior improves sparse performance.

Matched-layer diagnostic. We evaluate Full Learn-QK and SWA-hybrid Learn-QK after the same 64K continuation stage, using the same 50 tokenized 64K RULER prompts. For each prompt, we inspect the last query row of the *dense* prefill attention distribution. To compare corresponding layers, we restrict Full to indices $\{3, 7, 11, 15, 19, 23\}$, the six full-attention layers of the hybrid. Each aggregate in Table 10 averages 50 prompts, six layers, and 16 heads. Entropy is measured in nats; normalized entropy divides by the log of the available key count. Mass@k is the probability mass in the k largest entries of a dense attention row. All reported differences are hybrid minus Full, with mass differences in percentage points.

Even with learned gains in both models, the hybrid has 1.001 nats higher mean entropy and 6.86 percentage points lower mass@512 on corresponding layers. The direction is also consistent across inputs: the hybrid has higher mean matched-layer entropy on 46 of 50 paired prompts and lower mass@512 on 43 of 50. Table 11 shows that the hybrid has higher entropy at all six matched layers and lower mass@512 at five of them.

The learned gains are nontrivial: across matched layers, the mean per-channel product of query- and key-normalization weights is 2.138 in the hybrid and 2.357 in Full. This weight summary is not

Model, matched six layers	Entropy	Norm. entropy	Mass@32	Mass@128	Mass@512	Mass@2048
Full Learn-QK	5.336	0.482	60.75%	69.14%	78.04%	87.35%
SWA hybrid Learn-QK	6.337	0.572	43.59%	57.68%	71.18%	83.36%
Hybrid − Full	+1.001	+0.090	−17.16 pp	−11.46 pp	−6.86 pp	−3.99 pp

Table 10: 64K attention entropy and retained mass with learned QK gains.

Layer	Full H	Hybrid H	ΔH	Full mass@512	Hybrid mass@512	Δ mass@512
3	5.759	6.823	+1.063	73.60%	65.41%	−8.19 pp
7	4.486	5.782	+1.296	84.95%	73.17%	−11.77 pp
11	3.989	5.349	+1.360	91.24%	81.49%	−9.75 pp
15	4.656	5.557	+0.901	84.58%	80.29%	−4.29 pp
19	5.942	6.514	+0.572	73.83%	74.18%	+0.36 pp
23	7.183	7.997	+0.815	60.06%	52.53%	−7.53 pp

Table 11: Layerwise entropy and top-512 mass with learned QK gains.

an exact logit multiplier. Despite these learned gains, the hybrid retains higher entropy and lower top-k mass. Together with the sparse-performance improvement in Table 3, these results show that learned gains alone do not fully address the hybrid's concentration deficit in our experiments.

E Concentration and Sparse Approximation: Derivations

We analyze a single attention row with fixed logits and value vectors, comparing its scaled dense distribution with its own renormalized top-k approximation. The training-time effects are evaluated in Section 5.

Retained mass and approximation error. Let $\boldsymbol{a}$ be a dense attention distribution, $\mathcal{S}_k$ the indices of its k largest entries, and $M_k = \sum_{i\in\mathcal{S}_k} a_i > 0$. Its renormalized top-$k$ approximation has $a_i^{(k)} = a_i/M_k$ on $\mathcal{S}_k$ and zero elsewhere. Since $M_k \le 1$, the discrepancy is

$$\begin{aligned}\|\boldsymbol{a} - \boldsymbol{a}^{(k)}\|_1 &= \sum_{i\in\mathcal{S}_k}\left(\frac{a_i}{M_k} - a_i\right) + \sum_{i\notin\mathcal{S}_k} a_i \\ &= (1-M_k) + (1-M_k) = 2(1-M_k).\end{aligned} \tag{14}$$

For value vectors satisfying $\|\boldsymbol{v}_i\|_2 \le B$, the triangle inequality gives

$$\left\|\sum_i a_i\boldsymbol{v}_i - \sum_i a_i^{(k)}\boldsymbol{v}_i\right\|_2 \le \sum_i |a_i - a_i^{(k)}|\,\|\boldsymbol{v}_i\|_2 \le 2B(1-M_k). \tag{15}$$

Logit scale and entropy. For a fixed finite logit vector $\boldsymbol{z}\in\mathbb{R}^n$ and $g>0$, write $a_i(g) = \exp(gz_i)/Z(g)$, where $Z(g) = \sum_j \exp(gz_j)$, and let $\mu(g) = \sum_i a_i(g)z_i$. Then $a_i'(g) = a_i(g)(z_i - \mu(g))$, so $\mu'(g) = \mathrm{Var}_{\boldsymbol{a}(g)}[z]$. Using $\mathcal{H}(\boldsymbol{a}(g)) = \log Z(g) - g\mu(g)$ yields

$$\frac{\partial\mathcal{H}(\boldsymbol{a}(g))}{\partial g} = -g\ \mathrm{Var}_{\boldsymbol{a}(g)}[z] \le 0. \tag{16}$$

Logit scale and retained mass. Positive scaling preserves the ordering of logits, so the top-k set $\mathcal{S}_k$ is fixed, with ties resolved consistently. For $1 \le k < n$, let μ_{in} and μ_{out} be the conditional means of z under $\boldsymbol{a}(g)$ inside and outside this set. Since $\mu = M_k\mu_{\mathrm{in}} + (1-M_k)\mu_{\mathrm{out}}$, we obtain

$$\begin{aligned}\frac{\partial M_k(g)}{\partial g} &= \sum_{i\in\mathcal{S}_k} a_i(g)(z_i - \mu) \\ &= M_k(g)\big(1-M_k(g)\big)(\mu_{\mathrm{in}} - \mu_{\mathrm{out}}) \ge 0.\end{aligned} \tag{17}$$

The inequality follows because each selected logit is at least as large as each excluded logit. For $k = n$, retained mass is identically one.

Thus, increasing g lowers entropy, increases retained mass, and tightens the sparse-approximation output-error bound under the stated fixed-logit and fixed-value setting.

F When Does the Entropy Bias Emerge?

Table 12 reports selected checkpoints from Figure 1, using the same 50 fixed 16K prompts and six matched full-attention positions. The entropy gap is present at the earliest archived probe, at approximately 3.3% of pretraining: the SWA hybrid exceeds Full by 0.660 nats, while retaining 11.0 percentage points less mass in its top 128 keys. By the end of pretraining, both models have become more concentrated: entropy decreases by 0.800 nats in Full and 0.700 nats in SWA, while top-128 mass increases by 14.0 and 13.0 points, respectively. The final entropy gap is 0.760 nats.

Table 12: Entropy and top-128 mass throughout pretraining.

	Entropy (nats)			Top-128 mass (%)	
Pretraining progress	Full	SWA hybrid	Gap	Full	SWA hybrid
3.3%	6.596	7.256	0.660	51.6	40.6
16.3%	6.238	6.762	0.524	56.9	47.9
32.5%	6.063	6.745	0.681	60.0	48.6
100%	5.796	6.556	0.760	65.6	53.6

Attention becomes more concentrated in both models during pretraining, while the hybrid's entropy bias persists from the earliest measured stage to the final checkpoint.

F.1 Full-Layer Parameter Changes

We compare parameter changes from approximately 35.8% to 100% of pretraining for three models: Full at product scale 1.44, SWA at 1.44, and SWA trained with the 2.56 concentration prior. For each matrix W in the six shared full-attention layers, we measure $\|W_{\text{final}} - W_{\text{mid}}\|_F / \|W_{\text{mid}}\|_F$ and average over layers, where W_{mid} and W_{final} denote weights at these two stages. We also report the final Q/K spectral effective rank, defined as the exponential of the entropy of normalized singular values.

Table 13: Relative full-layer parameter changes and final Q/K effective ranks.

Model	W_Q change	W_K change	W_V change	W_O change	W_Q rank	W_K rank
Full 1.44	0.688	0.538	0.458	0.528	1076	321
SWA 1.44	0.735	0.558	0.486	0.539	1044	301
SWA 2.56	0.734	0.538	0.478	0.546	1064	306

The SWA hybrid does not exhibit smaller full-layer parameter movement: its relative Q update exceeds Full's at all six matched layer positions, and its mean Q/K/V/O changes are each at least as large. SWA trained at $g = 2.56$ has a similar relative Q change to SWA at $g = 1.44$ (0.734 versus 0.735), with slightly higher final Q/K effective ranks. Thus, the entropy differences coexist with comparable parameter movement. We next examine gradients on a fixed retrieval task.

F.2 Q/K Gradients on a Fixed Retrieval Task

We backpropagate the gold-answer-token loss on the same 50 fixed 16K NIAH prompts through checkpoints at approximately 35.8% and 100% of pretraining, without an optimizer step. At every matched full-attention layer, we divide the gradient Frobenius norm by the corresponding weight norm, then average over the six layers and prompts. Table 14 shows that the hybrid's Q and K

Table 14: Normalized answer-loss gradients, $10^3\|\nabla_W \mathcal{L}\|_F/\|W\|_F$.

Pretraining progress	Model	W_Q	W_K	W_O
35.8%	Full 1.44	0.472	0.679	0.597
	SWA 1.44	0.465	0.702	0.322
	SWA 2.56	0.604	0.801	0.425
100%	Full 1.44	0.658	0.937	0.838
	SWA 1.44	0.928	1.344	0.614
	SWA 2.56	0.655	0.962	0.471

gradients are comparable to Full's at the intermediate stage and *larger* by the end of training on this controlled long-range task. The output projection shows the opposite tendency.

At the intermediate stage, the SWA-to-Full ratios for Q and K are 0.99 and 1.03; their paired prompt-bootstrap intervals include zero difference. At the final checkpoint, SWA 1.44 has $1.41\times$ and $1.44\times$ the normalized Q and K answer gradients. The paired prompt-bootstrap 95% intervals for these final SWA-minus-Full differences are strictly positive for both matrices: $[0.084, 0.438] \times 10^{-3}$ for Q and $[0.157, 0.641] \times 10^{-3}$ for K. The hybrid's Q/K gradients on this retrieval probe are therefore not smaller than Full's on average. SWA trained at $g = 2.56$ has smaller final answer gradients than SWA at $g = 1.44$, alongside lower answer loss on the same prompts.

F.3 Head-Level Retrieval and Entropy

We examine the association between head-level retrieval and entropy using the 400 matched 16K NIAH prompts of Section 3.2. For each of the 96 matched full-attention heads, we average its entropy and gold-answer mass over the prompts, then call a head retrieval-active if its mean answer mass reaches 0.02. Table 15 separates active from other heads. Because active heads can be concentrated in particular layers, the last column first centers each head's entropy by its layer mean. We select active heads on one half of the prompts and compute their centered-entropy contrast on the other half, then reverse the split and average the two contrasts. The held-out contrast is active minus inactive centered entropy; negative values indicate more concentrated retrieval-active heads.

Table 15: Head-level retrieval and entropy on the NIAH probe.

Model	Active heads	Active entropy	Other entropy	Held-out contrast
Full	8/96	6.35	5.94	-0.05
SWA hybrid	14/96	6.41	6.70	-0.23
GDN hybrid	10/96	6.75	6.36	$+0.15$

SWA's retrieval-active heads are more concentrated than the remaining heads within the same layers in both held-out directions (-0.25 and -0.21 nats). GDN shows the opposite association ($+0.20$ and $+0.10$ nats). The relationship between retrieval activity and entropy therefore differs across architectures; the SWA model's higher overall entropy does not imply that its retrieval-active heads are more diffuse.

F.4 Training-Time Prior versus Post-hoc Scaling

To distinguish the immediate softmax effect of a larger QK scale from its effect on training, we evaluate 50 fixed 16K `niah_single_1` prompts under three settings: the SWA 1.44 checkpoint at its native scale, the *same* checkpoint with its six full layers temporarily changed to 2.56 for the entire forward pass, and the independently trained SWA 2.56 checkpoint at its native scale. We report the last prefill query at the six full layers. Answer mass is the mean attention probability on the annotated gold-answer tokens across their 96 heads; answer NLL is the mean loss on the actual answer tokens after the prompt, excluding the preceding whitespace. Far mass measures attention on keys more than 2,048 tokens behind the query.

Table 16: Post-hoc scaling versus training with the concentration prior.

Setting	Entropy	Top-128 (%)	Far >2K (%)	Gold (%)	NLL
SWA 1.44 (native)	5.63	65.9	39.2	3.05	0.262
SWA 1.44 ($g \to 2.56$)	2.89	93.0	18.6	0.93	0.565
SWA 2.56 (trained)	4.35	78.8	29.8	3.03	0.205

64K RULER task	Fixed $g = 1.44$	Fixed $g = 2.56$	Learn-QK	Learn-QK + $g = 1.96$
Mean (13 tasks)	26.57	35.67	28.13	32.92
Mean (NIAH)	36.56	50.31	37.94	46.31
Mean (non-NIAH)	10.57	12.24	12.43	11.48
Single-key 1	78.00	100.00	100.00	98.00
Single-key 2	46.00	72.00	44.00	80.00
Single-key 3	58.00	74.00	50.00	76.00
Multi-key 1	56.00	66.00	68.00	66.00
Multi-key 2	10.00	20.00	8.00	0.00
Multi-key 3	0.00	6.00	0.00	2.00
Multi-value	21.00	36.50	15.50	25.50
Multi-query	23.50	28.00	18.00	23.00
Variable tracking	0.00	13.60	19.60	21.20
Common words	0.20	1.60	1.20	0.20
Frequent words	6.67	4.00	1.33	0.00
QA-1	22.00	20.00	22.00	16.00
QA-2	24.00	22.00	18.00	20.00

Table 17: Per-task 64K RULER scores with trained top-2,048 indexers. NIAH denotes needle-in-a-haystack retrieval.

Post-hoc scaling sharply lowers entropy but reduces gold-answer mass from 3.05% to 0.93% and increases answer NLL from 0.262 to 0.565. Training with $g = 2.56$ instead increases top-128 mass from 65.9% to 78.8%, preserves gold-answer mass at 3.03%, and lowers NLL to 0.205. The paired prompt-bootstrap 95% intervals for the NLL changes are $[0.152, 0.495]$ for post-hoc scaling and $[-0.082, -0.029]$ for training with the prior. Thus, training with the concentration prior improves coverage while preserving useful retrieval evidence; simply sharpening a pretrained model does not reproduce this benefit.

The full-forward intervention also changes key rankings in later layers. In the native SWA model, at least one gold answer token lies among a head's top 128 keys for 58.4% of the 96 matched heads averaged over prompts. Post-hoc 2.56 reduces this to 34.2%; the trained-2.56 model retains 53.1%. At the first full layer (layer 3), the post-hoc switch leaves this rank statistic unchanged at 7.5%, as it must when only fixed logits are rescaled. By layer 7, however, gold top-128 inclusion falls from 62.4% to 26.3%. Sharpening an upstream full layer has changed the representations and the *ranking* at subsequent full layers, not just their softmax temperature. This suggests that adapting to these propagated changes is part of the benefit of training with the prior.

G Detailed Evaluation with Trained Sparse Indexers

Table 17 reports the per-task RULER scores, and Table 18 reports the category-level LongBench scores for the four trained-indexer checkpoints. Evaluation protocols are given in Appendix A.

LongBench v1 category	Fixed $g = 1.44$	Fixed $g = 2.56$	Learn-QK	Learn-QK + $g = 1.96$
Mean (21 tasks)	21.74	21.81	20.54	20.91
Single-document QA	12.05	12.09	10.88	10.66
Multi-document QA	9.50	9.91	9.21	9.28
Summarization	16.90	17.95	17.73	17.32
Few-shot learning	46.72	47.31	43.48	45.29
Synthetic	3.43	2.49	4.01	2.58
Code	52.73	50.69	47.09	50.59

Table 18: LongBench scores by task category with trained top-2,048 indexers.