WHEN CAN ATTENTION HEADS BE STATICALLY DEFINED?

Anonymous authors
Paper under double-blind review

ABSTRACT

Some attention heads learn similar patterns across inputs. Reusing these patterns could reduce training cost by avoiding repeated query-key score computation and softmax. Through controlled pretraining comparisons, we identify Selective Attention Freezing (SAF), which selects heads with low attention-pattern variance and replaces their attention weights with fitted post-softmax means halfway through training. We represent these fixed patterns with absolute-position and relative-distance preferences, reducing storage from quadratic to linear in sequence length. A fused kernel reconstructs the patterns and executes ordinary-attention and replaced heads together. At matched training-token budgets, replacing 25% of attention heads gives 1.056× faster post-replacement optimiser updates at 124M parameters and 4K context, with a 0.77% perplexity increase. At 1B and 8K context, post-replacement updates are 1.068× faster on four GPUs including communication, with a 0.51% perplexity increase. The resulting models also accelerate long-input finetuning and causal prefill. After associative-recall adaptation, the 124M model with 25% replacement generalises to more key-value pairs at a fixed length better than ordinary attention and two pruning controls.

1 INTRODUCTION

Self-attention computes its attention weights for every head and every input (Vaswani et al., 2017). For each head, query-key interactions determine an input-dependent attention matrix, allowing the model to dynamically decide which tokens should interact. This input-dependence is the source of the mechanism's expressivity, but it comes at a computational cost: each head requires query and key projections, attention-score computation, and softmax normalisation, together with their backward passes during training, giving quadratic cost in sequence length.

For some heads, however, the attention weights depend primarily on query and key positions rather than on the tokens themselves, making their scores nearly content-free. Positional encodings make such heads easy to form, as shown for RoPE (Barbero et al., 2025), and fixed positional patterns can replace learned heads with little loss in translation and pretrained encoders (Raganato et al., 2020; Tay et al., 2021; Hassid et al., 2022). Recent language-model studies also examine relaxed sequence dependence and random input-independent token mixing (Xue et al., 2025; Dong et al., 2025). For a head whose scores no longer depend on content, computing them for every input reproduces the same positional bias while still paying for the computational cost.

This observation means we can replace such heads during training, but existing results remains unclear which heads and patterns to choose, when to replace them, and if a layer that mixes fixed and ordinary-attention heads yields memory and wall-clock savings with modern kernels. So we ask:

(1) Under what conditions can an input-dependent attention matrix be replaced by a fixed pattern and (2) how does this improve computational and memory efficiency?

Our controlled comparisons lead to Selective Attention Freezing (SAF), a recipe that replaces selected attention matrices with fixed causal patterns. SAF fixes the attention weights but retains token mixing of input-dependent values: the value ($\mathbf{V}$) and output projections remain trainable. The hybrid layer combines ordinary query-key attention with fixed-pattern heads (Figure 1). We compare patterns, selectors, rates, and timing during pretraining, including pruning at matched tokens and time, then evaluate finetuning and causal prefill.

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.

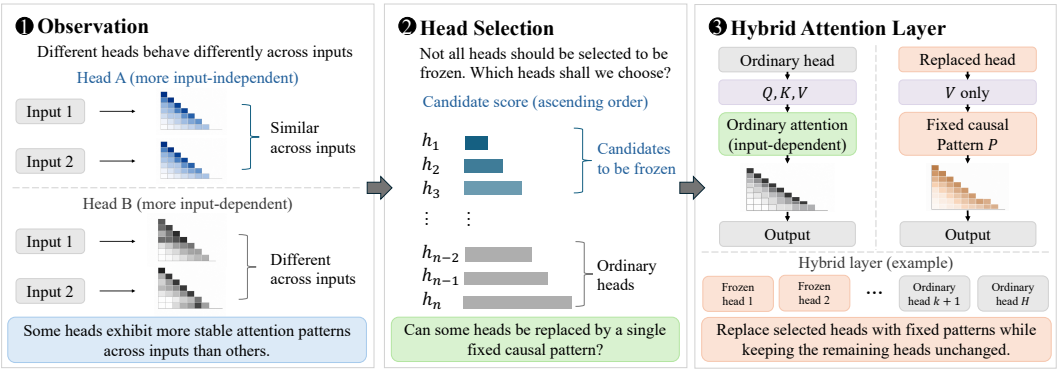

Figure 1: Overview of fixed-pattern replacement of ordinary attention.

A dense fixed attention matrix requires quadratic space. Because a content-free pattern is a positional bias, we store it as absolute-key-position and relative-distance preferences in vectors of linear size, and develop a fused kernel that executes ordinary attention with FlashAttention (Dao et al., 2022) and frozen heads in one forward pass. Reconstructing fixed patterns in registers bypasses softmax, and **Q/K** gradients in our multi-head pretraining path, reducing pattern bandwidth and activation storage.

At 25% replacement, SAF increases perplexity by $0.77 \pm 0.07\%$ at 124M and 4K context, with $1.056\times$ faster post-replacement updates. At 1B and 8K, the increase is 0.51% with a $1.068\times$ speedup on four GPUs, including communication. These savings accumulate over the remaining training updates, reducing the accelerator time needed to process a fixed token budget. The 50% setting offers larger speed and memory gains with a higher perplexity cost. Longer-context 124M models have smaller perplexity penalties and accelerate long-input finetuning and causal prefill. After adaptation on eight key-value pairs, the 124M SAF models outperform both pruning baselines on 24–64 pairs.

2 Related Work

Input-independent and constrained attention. Fixed positional patterns in translation (Raganato et al., 2020; You et al., 2020), input-independent audio encoders (Wu et al., 2020), and learned or random synthetic token mixing (Tay et al., 2021; Dong et al., 2025) show that useful computation can persist without standard attention. PAPA uses input-averaged attention in pretrained encoders (Hassid et al., 2022), and recent work relaxes sequence dependence in language models (Xue et al., 2025). Theory also studies expressivity under frozen weights or sparse graphs (Zaheer et al., 2020; Fu et al., 2023; Otsuka et al., 2025). We build on input-averaged attention through selective replacement during causal language-model training, compact storage, and fused execution.

Head heterogeneity and efficient training. Distinct head functions motivate pruning and component-importance measures (Michel et al., 2019; Voita et al., 2019; Men et al., 2025). FLAP uses activation fluctuations to guide structured pruning and compensates removed features with a constant bias (An et al., 2024). Fixed-pattern replacement instead retains token mixing of the current input's values. Retrieval-head analysis studies key-value recall (Wu et al., 2025); DuoAttention separates retrieval and streaming heads (Xiao et al., 2025), while MInference assigns head-specific sparse patterns (Jiang et al., 2024). Related training strategies freeze or prune components progressively (Brock et al., 2017; Zhang & He, 2020; Liu et al., 2021; Xia et al., 2024; Erdogan et al., 2025), and quantisation-aware training also exposes compute-allocation trade-offs (Dremov et al., 2026).

Hardware-aware execution. FlashAttention provides efficient exact attention (Dao et al., 2022; Dao, 2024), while MInference and DuoAttention exploit head heterogeneity within kernels (Jiang et al., 2024; Xiao et al., 2025). Our kernel executes ordinary attention and fixed-pattern token mixing together, removing score computation and softmax for replaced heads. The pretraining path also omits their query and key projections; projection savings depend on whether keys are shared (Section 3.3).

3 REPLACING ATTENTION WITH A FIXED PATTERN

After an initial period of training with ordinary attention, we use a set of calibration inputs to estimate attention statistics at the current checkpoint. We then select heads, construct fixed patterns, and resume training. Algorithm 1 gives the resulting one-time SAF procedure.

Let L denote the number of attention layers and H the number of query heads per layer, giving LH heads in total. We use T for sequence length and d_h for head dimension. For normalised layer input $\mathbf{X}_\ell$, let $\mathbf{Q}_{\ell h}, \mathbf{K}_{\ell h}, \mathbf{V}_{\ell h}$ be the head projections and $\mathbf{S}_{\ell h} = \mathbf{Q}_{\ell h}\mathbf{K}_{\ell h}^\top/\sqrt{d_h}$ the attention scores. Ordinary attention computes input-dependent query–key scores and mixes values using $\mathbf{A}_{\ell h}(\mathbf{X}_\ell) = \mathrm{softmax}_{\mathrm{causal}}(\mathbf{S}_{\ell h})$. With i, j indexing query and key positions, we replace a selected head's attention weights $\mathbf{A}_{\ell h}$ by a fixed causal row-stochastic pattern $\mathbf{P}_{\ell h} \in \mathbb{R}^{T\times T}$, giving the head output: $\widetilde{\mathbf{Z}}_{\ell h}(\mathbf{X}_\ell) = \mathbf{P}_{\ell h}\mathbf{V}_{\ell h}$.

Both paths perform token mixing: a weighted sum of value vectors across token positions. Value and output projections remain trainable, so the head output still depends on the input through $\mathbf{V}_{\ell h}$; only its attention weights are fixed. For replaced heads in multi-head attention, query and key projections, score computation, and softmax are removed. Appendix A.1 gives full tensor definitions.

3.1 CALIBRATION AND HEAD SELECTION

Attention heads exhibit different patterns and roles (Michel et al., 2019; Voita et al., 2019; Clark et al., 2019b; Li et al., 2023; Neo et al., 2024; Jiang et al., 2024; Li et al., 2026). We compare scores based on attention variation and head outputs. At the replacement checkpoint, we measure attention on N calibration sequences in evaluation mode, leaving model weights and the training-data sampler unchanged. For calibration input $\mathbf{X}_{\ell,n}$, write $\mathbf{A}_{\ell h}^{(n)} = \mathbf{A}_{\ell h}(\mathbf{X}_{\ell,n})$ and $\mathbf{S}_{\ell h}^{(n)} = \mathbf{S}_{\ell h}(\mathbf{X}_{\ell,n})$; the empirical mean is $\widehat{\mathbf{A}}_{\ell h} = N^{-1}\sum_n \mathbf{A}_{\ell h}^{(n)}$. We rank heads globally by increasing score $s_{\ell h}$, targeting $k = \mathrm{round}(rLH)$ heads at replacement rate r. The attention-variance score is

$$s_{\ell h}^{\mathrm{var}} = \frac{1}{(N-1)T^2}\sum_{n=1}^{N}\left\|\mathbf{A}_{\ell h}^{(n)} - \widehat{\mathbf{A}}_{\ell h}\right\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm. Calibration data are disjoint from reporting data (Appendix A.1). The variance score measures reconstruction error across inputs, rather than functional importance, and also depends on attention concentration. The forward-KL score instead compares each causal query-row distribution with its empirical mean:

$$s_{\ell h}^{\mathrm{KL}} = \frac{1}{NT}\sum_{n=1}^{N}\sum_{i=1}^{T}\mathrm{KL}\Big(\mathbf{A}_{\ell h}^{(n)}[i,1{:}i]\,\Big\|\,\widehat{\mathbf{A}}_{\ell h}[i,1{:}i]\Big). \tag{2}$$

It measures input dependence in probability space, accumulating directly from row entropies during the same calibration pass. Appendix A.3 introduces alternative selection metrics, including Q/K-gradient scores, output magnitude, within-layer redundancy, and residual influence. Appendix B.2 evaluates these alternatives by measuring the immediate loss penalty of single-head replacements and the long-term model perplexity after continued training.

For compact patterns (Section 3.2), we consider candidates in score order and skip fits whose row-averaged KL error exceeds $\varepsilon_{\mathrm{fit}}$. Selection continues until k fits pass or no candidates remain. The score determines candidate order; $\varepsilon_{\mathrm{fit}}$ bounds compact-fitting error.

3.2 FIXED-PATTERN CONSTRUCTION AND FITTING

For each candidate head, we compare five causal row-stochastic patterns: deterministic averages, fitted distributions, and a structured-random control.

- **Post-softmax mean.** $\mathbf{P}_{\ell h} = \widehat{\mathbf{A}}_{\ell h}$ averages attention probabilities across inputs and remains causal and normalised.
- **Sharp mean.** $\mathbf{P}_{\ell h} = \mathrm{softmax}_{\mathrm{causal}}(N^{-1}\sum_n \mathbf{S}_{\ell h}^{(n)})$ averages logits before softmax.

- **Gaussian sample.** We sample causal logits independently from normal distributions fitted to their means and variances across inputs, then apply causal softmax. This captures marginal logit variation but discards correlations.
- **Dirichlet sample.** We sample each causal row jointly with mean $\widehat{\mathbf{A}}_{\ell h}[i, 1{:}i]$ and concentration fitted from across-input variances. The draws are nonnegative and normalised without softmax.
- **Structured random (control).** We sample $\alpha_{\ell h}(j), \rho_{\ell h}(\delta) \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$ and construct Equation 3, testing random preferences within the same positional parameterisation.

Each sampled pattern is drawn once at replacement and retained thereafter. Appendix C.2 gives distribution-fitting details and empirical checks.

Compact representation. Dense patterns require $O(T^2)$ storage per head. We represent absolute-key-position and relative-distance preferences with length-T vectors α and ρ, motivated by distance structure and attention sinks (Xiao et al., 2024; Barbero et al., 2025; Yang et al., 2026). Suppressing head indices, α uses key indices $1, \ldots, T$ and ρ uses offsets $0, \ldots, T-1$. With $Z(i) = \sum_{k=1}^{i} \exp(\alpha(k) + \rho(i-k))$, the pattern is

$$\widehat{\mathbf{P}}(i,j) = \begin{cases} \dfrac{\exp(\alpha(j) + \rho(i-j))}{Z(i)}, & j \le i, \\ 0, & j > i. \end{cases} \tag{3}$$

Each query row is normalised separately. The vectors α and ρ capture absolute-position effects such as attention sinks and query-key distance, respectively, using $O(T)$ storage per head.

Fitting. We fit the compact pattern in Equation 3 to a target fixed pattern $\mathbf{P}$ by minimising row-averaged cross-entropy:

$$\mathcal{L}_{\text{fit}}(\alpha, \rho) = \frac{1}{T}\Big[\sum_i \log Z(i) - \sum_j \eta_{\text{abs}}(j)\,\alpha(j) - \sum_\delta \eta_{\text{rel}}(\delta)\,\rho(\delta)\Big], \tag{4}$$

where $\eta_{\text{abs}}(j) = \sum_{i \ge j} \mathbf{P}(i,j)$ and $\eta_{\text{rel}}(\delta) = \sum_{i-j=\delta} \mathbf{P}(i,j)$ are the target's absolute-position and relative-distance marginals. These two length-T marginals are sufficient statistics for fitting. Their linearity permits online accumulation during calibration, although our implementation first collects dense attention statistics. We use a fitting tolerance of $\varepsilon_{\text{fit}} = 0.2$ nats per row. Comparisons with prescribed head sets require every fit to pass, without substituting other heads (Appendix A.2). Appendix D gives the optimality conditions and memory-bounded fitter.

3.3 Replacement schedules and fused execution

We store the fixed patterns as buffers and resume training from the same model weights and optimiser state. Ordinary-attention heads and the remaining parameters, including value and output projections, continue to train. α, ρ, and the normalisers receive no gradients.

Replacement schedules. Gradual and iterative pruning strategies motivate varying both the extent and timing of replacement (Xia et al., 2024; Shen et al., 2024; Wang et al., 2026). One-time replacement freezes the entire selected set at one update. Gradual replacement installs the same final heads in nearly equal groups at evenly spaced updates, testing whether smaller interventions ease adaptation. The validation-budget schedule controls cumulative measured validation-perplexity cost, while the train-loss-triggered schedule uses training-loss plateaux to propose replacements and waits for recovery between groups. Appendix A.3 formalises these schedules and details their settings.

Fused computation. Our fused kernel executes ordinary-attention and replaced heads in one forward launch per layer. A head-state flag selects each kernel program's computation, avoiding separate launches and concatenation. Ordinary-attention heads use FlashAttention-2's online-softmax tiling (Dao et al., 2022; Dao, 2024). Replaced heads reconstruct causal tiles of $\widehat{\mathbf{P}}$ in registers from α, ρ, and precomputed normalisers, then multiply them with value tiles (Figure 2). This avoids score computation, softmax, and dense pattern reads for replaced heads. Their attention backward pass is $\mathrm{d}\mathbf{V} = \widehat{\mathbf{P}}^\top \mathrm{d}\mathbf{Z}$. Our multi-head pretraining implementation omits replaced query and key projections;

the Qwen grouped-query path omits replaced query projections but retains full shared key and value projections. Storage is $O(T)$ per replaced head, while token mixing remains $O(T^2 d_h)$. Appendix D gives execution details and numerical validation; Section 5.3 measure update and prefill costs.

4 Experimental Setup

Model pretraining. We train a 124M decoder with 12 Transformer layers, 12 heads per layer, and hidden dimension 768 on FineWeb-Edu (Penedo et al., 2024), comparing learned absolute positions with RoPE (Su et al., 2024) at 4K context. Each run performs 5,000 updates of 491,520 target tokens, totalling 2.4576B tokens or approximately 20 tokens per parameter (Hoffmann et al., 2022). Replacement runs share the source checkpoint, training configuration, and subsequent data order. Pattern comparisons use identical head sets; rate comparisons use nested subsets of one ranking. Replacement occurs at the midpoint unless timing is varied. Gaussian, Dirichlet, and sharp patterns use dense storage in the pattern comparison. The post-softmax mean is tested in dense and compact forms; systems measurements use the compact representation. At 4K, calibration uses 32 sequences from the training corpus, sampled with a separate random-number generator so that the training-data order is unchanged. Calibration inputs are disjoint from final reporting data (Appendix A.1).

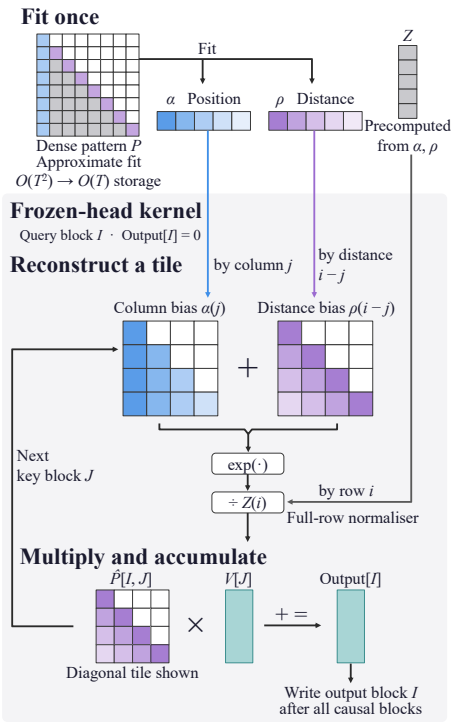

Figure 2: Compact representation and execution of a fixed attention pattern.

Control baselines. Three-seed controls compare pruning and random head selection with mean replacement at matched tokens. We use micro-batch size 8, 15 accumulation steps, and shuffled non-overlapping training windows. Random selection retains fitted means and matches per-layer head counts. These controls use 32 held-out calibration sequences (Appendix C.3). At matched time, continuations share an allowance and elapsed-time learning-rate schedule, so faster models process more tokens. We also compare frozen and trainable compact mean patterns from identical fitted initialisations, and evaluate both in multi-query associative recall (MQAR; Appendix C.3.2; Arora et al., 2024). Gate-Taylor pruning scores each head by the mean absolute loss gradient with respect to a scalar gate on its output (Michel et al., 2019). We normalise scores within each layer and remove the globally lowest-scoring heads once (Appendix C.3.4).

Quality and runtime. For replaced and ordinary-attention held-out perplexities p_{rep} and p_{ord}, we report $\Delta\mathrm{PPL} = 100(p_{\mathrm{rep}}/p_{\mathrm{ord}} - 1)$. We compare ordinary-attention and replaced checkpoints on SST-2 (Socher et al., 2013), BoolQ (Clark et al., 2019a), and QuALITY (Pang et al., 2022) over three finetuning seeds each. Pruning and MQAR (Arora et al., 2024) comparisons vary pretraining and task-training seeds separately (Appendices C.3.4 and C.3.6). All systems measurements use GH200 GPUs with bfloat16 computation. Pretraining update timings include gradient accumulation, backward computation, clipping, AdamW, and gradient reset, but exclude data loading, calibration, scoring, and fitting. The supplementary controls report total training times including the initial training with ordinary attention and intervention. Speedups aggregate within-pair ratios of ordinary-attention to replaced-model update times; displayed times are aggregated separately, using medians for single-GPU pretraining benchmarks and means for finetuning and four-GPU benchmarks. Quality and runtime use separate runs in the 124M study (Appendices C.1 and C.4).

Transfer evaluations. The 8K and 16K runs retain the 2.4576B-token budget, training models and fitting patterns separately at each length. Their checkpoints provide square causal-prefill benchmarks across lengths and batch sizes, excluding token-by-token KV-cache decoding. Qwen3-4B is evaluated

on five zero-shot tasks without further training; variance and forward-KL selection each use 512 calibration sequences (Appendix C.6). At 1B parameters and 8K context, ordinary-attention, replaced, and Gate-Taylor-pruned models are applied on the same midpoint checkpoint, and a 19.667B-token budget on four GH200 GPUs. We compare matched held-out PPL and three-seed finetuning, and benchmark ordinary-attention and replaced updates with four-GPU communication (Appendix C.5).

5 RESULTS

5.1 WHICH FIXED PATTERNS AND HEADS SHOULD BE USED?

Fixed-pattern content. At matched tokens, the post-softmax mean gives the lowest or similar perplexity across replacement rates, followed most closely by the sharp mean (Figure 3). Differences are small up to 25% replacement but widen at higher rates; neither Gaussian nor Dirichlet sampling improves on the mean. Appendices C.2 and B give diagnostics and reconstruction analysis.

Head selection. Variance selection gives lower perplexity after continued training than residual cosine, although residual cosine better predicts the immediate cost of replacing one head (Appendix B.2). Projected-output magnitude and redundancy are weaker predictors of this local cost (Table 5). Forward KL gives a larger perplexity penalty after adaptation at every tested context and rate (Table 6). We therefore retain attention variance for replacement during pretraining. At 25% replacement, selected heads concentrate in earlier layers and have more diffuse attention than unselected heads in those layers. Across three seeds, 30–31 of the 36 heads selected at quarter-training remain selected at the midpoint (Appendix B.2). With layer distributions matched, selected heads have higher mean normalised entropy (0.837 versus 0.678) and place less attention on the most recent 64 tokens (19.0% versus 40.7%). Low variance therefore favours diffuse attention, but its association with immediate replacement cost remains positive after controlling for both layer and entropy (partial Spearman correlation 0.543). Appendix B.2 gives the profile and one-head diagnostic protocols.

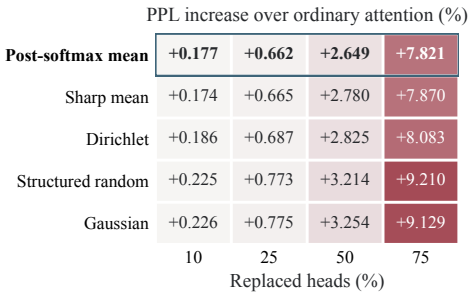

Figure 3: Perplexity increases for five fixed patterns and four replacement rates.

Head selection and retained token mixing. Random selection nearly doubles the 25% perplexity penalty despite matching per-layer head counts (Table 1). Variance selection also lowers the matched-time penalty from 1.24% to 0.46%, showing that its benefit extends beyond allocating more fixed heads to earlier layers.

Continuing to optimise fitted α, ρ reduces perplexity by only $0.010 \pm 0.006\%$ relative to freezing, while increasing update time by $11.9 \pm 1.2\%$ at 25% replacement (Table 14a). Keeping the fitted patterns fixed therefore gives nearly the same LLM performance with cheaper updates.

Retaining token mixing with the mean pattern gives lower perplexity than pruning the same low-variance heads at matched tokens. At matched time, pruning completes more updates and has lower perplexity than replacement. After longer training with ordinary attention, retaining token mixing still improves on same-head pruning across three seeds, but Gate-Taylor pruning gives lower perplexity under both budgets (Appendix C.3.4). Low variance therefore identifies heads whose weights can be fixed, rather than a general ranking of heads to remove.

5.2 HOW MANY HEADS SHOULD BE REPLACED, AND WHEN?

Replacement rate. Replacement rate has the largest effect on model quality. Across three pretraining seeds, midpoint replacement by the post-softmax mean increases perplexity by $0.768 \pm 0.067\%$ at 25% and $2.492 \pm 0.083\%$ at 50% (mean $\pm$ standard error; Table 2a). We therefore prefer 25% for retaining quality and use 50% to examine the trade-off at a higher rate.

Table 1: Perplexity increase (%, lower is better) at matched tokens or time: mean ± SE over three seeds. Times include the initial training with ordinary attention and intervention.

	Matched tokens: 2.4576B				Matched time	
	ΔPPL (%)		Time (min)		ΔPPL (%)	
Operation + selection	25%	50%	25%	50%	25%	50%
Mean + variance	0.663 ± 0.018	2.495 ± 0.024	184.7	178.7	0.456 ± 0.108	1.911 ± 0.112
Pruning + variance	0.806 ± 0.018	3.033 ± 0.056	178.9	168.3	0.113 ± 0.169	0.978 ± 0.108
Mean + random heads	1.282 ± 0.056	3.630 ± 0.093	183.4	178.8	1.243 ± 0.112	2.778 ± 0.125

Ordinary-attention PPL: 23.896 at matched tokens (188.7 min); 23.882 at matched time (187.65 min). Appendix C.3 gives absolute PPL, update counts, and paired contrasts.

Table 2: Quality and training costs for ordinary-attention and SAF 124M checkpoints. Ord./repl. (ms) reports optimiser-update times for the ordinary-attention and replaced models.

(a) Pretraining on FineWeb-Edu		Perplexity ↓			Update time		Memory
Context (B)	Replaced heads	Ordinary attn.	Replaced	ΔPPL (%)	Ord./repl. (ms)	Speedup	Peak change
4K (8)	25%	23.862	24.045	$+0.768 \pm 0.067\%$	2249.49/2129.42	1.056×	−1.96%
4K (8)	50%	23.862	24.457	$+2.492 \pm 0.083\%$	2229.80/1991.85	1.119×	−3.04%
8K (4)	25%	24.179	24.318	$+0.578 \pm 0.011\%$	2710.24/2636.22	1.030×	−1.63%
8K (4)	50%	24.179	24.676	$+2.058 \pm 0.066\%$	2782.97/2524.39	1.101×	−2.55%
16K (2)	25%	24.428	24.538	$+0.450 \pm 0.012\%$	3741.42/3649.78	1.025×	−0.82%
16K (2)	50%	24.428	24.865	$+1.792 \pm 0.112\%$	3697.14/3454.31	1.070×	−1.57%

(b) Supervised finetuning		Accuracy (%) ↑			Update time		Memory
Task (B)	Replaced heads	Ordinary attn.	Replaced	Δaccuracy (pp)	Ord./repl. (ms)	Speedup	Peak change
SST-2 (256)	25%	86.39	86.54	$+0.15 \pm 0.53$ pp	50.11/58.22	0.901×	−3.65%
SST-2 (256)	50%	86.39	86.77	$+0.38 \pm 1.00$ pp	50.11/56.70	0.925×	−5.95%
BoolQ (64)	25%	71.40	70.62	-0.77 ± 0.56 pp	67.36/70.91	0.967×	−4.63%
BoolQ (64)	50%	71.40	70.84	-0.56 ± 0.70 pp	67.36/69.35	0.992×	−7.19%
QuALITY 16K (4)	25%	26.59	26.09	-0.50 ± 0.63 pp	578.72/552.90	1.047×	−2.12%
QuALITY 16K (4)	50%	26.59	26.94	$+0.35 \pm 0.18$ pp	578.72/517.88	1.118×	−3.92%

Replacement time. Replacement is robust across a broad range of training stages, with the largest penalties when little or no training remains (Table 8 in Appendix C.1). The penalty is 0.662–0.724% for replacement between 25% and 60% of training, rising to 0.936% at 75% and 8.168% after training. Continued training enables adaptation, with little sensitivity to the exact update across earlier stages. Across three seeds, moving 25% replacement from the midpoint to quarter-training raises the perplexity penalty from 0.663% to 0.772% and saves 100 seconds on average (Appendix C.3).

The post-softmax mean outperforms structured-random patterns at every tested replacement time and rate, retaining its advantage after adaptation.

Fixed and automatic schedules. Neither gradual nor automatic replacement improves the quality–speed trade-off over one-time replacement (Tables 9 and 10 in Appendix C.1). At matched 25% and 50% rates, the train-loss-triggered schedule achieves similar quality across three seeds but takes longer because it repeatedly measures attention. Validation-loss budgets provide direct quality control but select substantially fewer heads under the tested budgets. We use one-time midpoint replacement: these fixed-token comparisons show low perplexity penalties with a single measurement.

The SAF recipe. These comparisons leads to SAF: variance selection, compact mean patterns, and one-time midpoint replacement, with 25% frozen rate. The mean minimises fixed-matrix reconstruction error, and variance measures this minimum for each head (Appendix B), motivating both choices. Compact fitting changes final perplexity by less than 0.014% relative to the dense mean under both position encodings; Appendix D reports agreement with the numerical reference.

5.3 Does the trade-off carry across model use stages?

Pretraining time and memory. Post-replacement update speed and memory savings increase with the number of replaced heads (Figure 4a). Updates are faster at every batch size from 2 to 8. Table 1 reports full training-path times, including the initial training with ordinary attention and intervention.

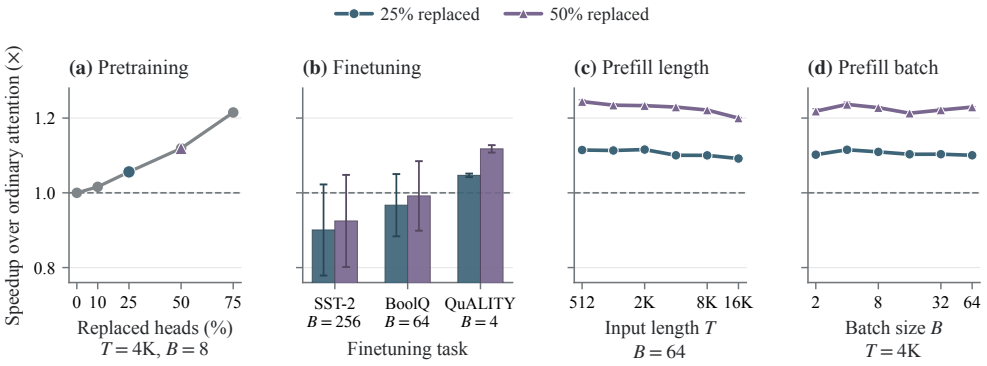

Figure 4: SAF speedup over FlashAttention: (a) pretraining updates; (b) finetuning updates; (c) causal prefill by input length at $B = 64$; (d) prefill by batch size at $T = 4096$.

Longer pretraining contexts. Without retuning, the perplexity penalty decreases from 4K to 16K at both rates (Table 2a). Updates remain faster and use less peak memory at 8K and 16K, although their measured speedups are smaller than at 4K. SAF is applied with models trained and patterns fitted separately at each context length.

Downstream adaptation. Across SST-2, BoolQ, and QuALITY, every absolute mean accuracy change is below 0.8 percentage points, with variation measured over three finetuning seeds (Table 2b). QuALITY provides a long-input finetuning workload, although accuracy with ordinary attention is close to the four-choice baseline. Finetuning speed depends on input length: SST-2 slows down, BoolQ approaches parity, and long-input QuALITY reaches 1.118× faster updates at 50% replacement (Figure 4b). Replacing heads during finetuning does not improve the quality–speed trade-off; Appendix C.4 reports the results and controller costs.

Associative recall. Retaining fixed-pattern token mixing improves generalisation to more associations within the same context. MQAR requires retrieving values paired with earlier keys (Arora et al., 2024). All four 124M models adapt on 512-token sequences with 8 pairs, reaching over 99.5% mean accuracy. At 64 pairs, SAF retains 54.4% accuracy versus 26.2–29.0% for ordinary attention and pruning (Figure 5). It outperforms pruning controls in every seed at 24, 32, 48, and 64 pairs. Length-only tests favour pruning, and immediate replacement in a task-trained ordinary-attention model does not give the same consistent benefit (Appendix C.3.6). In the midpoint, matched-token controls, frozen mean also gives higher mean accuracy than ordinary attention at 16–64 pairs (Table 14b).

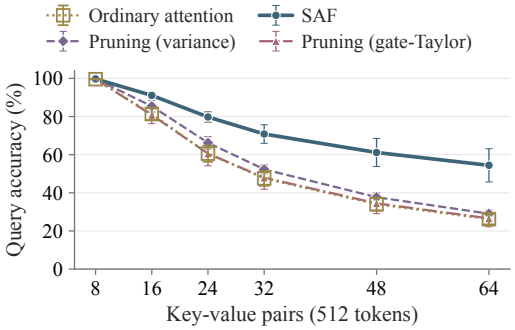

Figure 5: MQAR accuracy for later-intervention, matched-time 124M checkpoints. Models train on eight pairs at 512 tokens. (The ordinary-attention and gate-Taylor pruning curves nearly overlap.)

At a fixed input length, increasing the number of key-value pairs tests generalisation to more associations rather than to longer sequences. The higher MQAR accuracy indicates that these heads remain useful for retrieving key-value associations even when their attention weights are fixed.

Causal prefill. For the 124M 16K RoPE checkpoints, 25% and 50% replacement accelerate causal prefill by 1.09–1.12× and 1.20–1.24× across input lengths at $B = 64$. Prefill is also faster across the 4K batch-size sweep (Figure 4c,d). Appendix D gives absolute latencies and batch-one measurements, where launch overhead can outweigh the savings. The computational benefit extends

beyond pretraining: the same fixed patterns support faster long-input finetuning and causal prefill without being fitted again for these stages.

Zero-shot task accuracy in Qwen3-4B. Without further training, variance selection gives higher accuracy than forward KL on all tasks at 10%, 20%, and 30% replacement (Figure 6). At 10%, accuracy changes range from -1.13 to $+0.33$ percentage points relative to ordinary attention, averaging -0.37 points. Higher replacement rates give larger mean accuracy losses. Appendix C.6 reports per-task accuracies, selector analysis, and a separate GQA prefill benchmark.

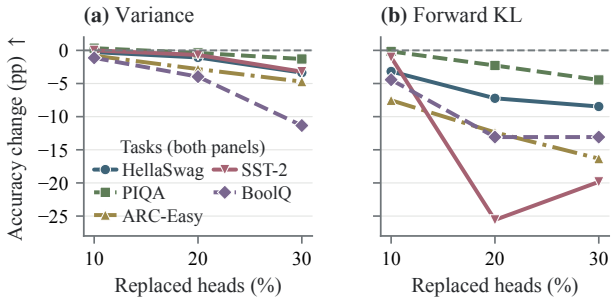

Figure 6: Qwen3-4B zero-shot accuracy changes (pp) relative to the ordinary-attention model.

5.4 Does SAF transfer to a larger model?

At 1B and matched 19.667B training tokens, 25% replacement increases perplexity by 0.507% and changes mean finetuning accuracy by less than 0.72 percentage points per task (Table 3). At 50%, perplexity increases by 2.027% with larger update savings. At 25%, SAF has slightly lower perplexity than Gate-Taylor pruning; mean task accuracies differ by less than 0.20 points, with no consistent advantage across finetuning seeds. These comparisons use one pretraining seed and three finetuning seeds; QuALITY accuracy remains close to the four-choice baseline.

Table 3: 1B quality at 8K context and 19.667B training tokens. PPL uses one pretraining seed; task accuracies (%) show mean ± SE over three finetuning seeds.

		Finetuning accuracy (%) ↑		
Model	PPL ↓	SST-2	BoolQ	QuALITY
Ordinary attention	12.446	91.44 ± 0.25	75.98 ± 0.24	27.44 ± 1.25
SAF 25%	12.509	91.63 ± 0.35	75.27 ± 1.45	27.60 ± 0.21
Pruning 25% (gate-Taylor)	12.529	91.70 ± 0.34	75.46 ± 1.19	27.66 ± 0.71
SAF 50%	12.698	90.98 ± 0.43	75.68 ± 0.31	25.12 ± 0.75

With distributed communication on four GH200 GPUs, the 25% and 50% checkpoints give update speedups of 1.068× and 1.167×, respectively. These post-replacement measurements include gradient accumulation, optimisation, and all-reduce; Appendix C.5 reports separate single-GPU memory and finetuning-time measurements. The 25% setting reduces update time by 6.4%, so applying this measured saving to a post-replacement budget of one million GPU-hours would save approximately 64,000 GPU-hours. With midpoint replacement, the corresponding estimate is 3.2% of the full update budget before the one-off intervention cost; the absolute saving grows with the training budget.

6 Conclusion

We have studied when attention heads can use fixed causal patterns during language-model training. Controlled comparisons identify Selective Attention Freezing (SAF): variance selection and fitted mean patterns with compact storage and fused execution. At 124M, freezing fitted patterns gives nearly the same perplexity as continued pattern learning, with faster updates. At 25% replacement, perplexity increases remain below 1% at matched tokens, with post-replacement updates 1.056× faster at 124M and 4K, and 1.068× faster at 1B and 8K on four GPUs. The models also support faster long-input finetuning and causal prefill. At 124M, retaining token mixing improves matched-token perplexity over same-head pruning. After associative-recall adaptation, it also improves generalisation to more key-value pairs at a fixed length over both pruning controls. These findings give a practical recipe for reducing input-dependent attention computation while retaining useful token mixing.

AI USE STATEMENT

In this work, we used generative AI tools to assist with method implementation and to refine experiments initially designed by the authors. We have not used generative AI tools to formulate mathematical claims, develop or write proofs, propose or refine hypotheses, interpret results, or determine the manuscript structure. Additionally, we used generative AI tools for code development and debugging, polishing text, and preparing result plots. We take responsibility for the final content of this work, including text, code, analyses and claims.

REPRODUCIBILITY STATEMENT

Section 3 and Appendices A–B detail the methods, assumptions, and derivations. Section 4 and Appendices C–D provide the experimental configurations, seeds, evaluation and measurement protocols, and run records. Appendix E lists datasets, pretrained models, and licences. The codes, data split, and model checkpoints will be made publicly available upon acceptance.

REFERENCES

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i10.28960. URL https://doi.org/10.1609/aaai.v38i10.28960.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Re. Zoology: Measuring and improving recall in efficient language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LY3ukUANko.

Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6(58):1705–1749, 2005. URL http://jmlr.org/papers/v6/banerjee05b.html.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes rotary positional encodings useful? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GtvuNrk58a.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019. URL https://arxiv.org/abs/1911.11641.

Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. Freezeout: Accelerate training by progressively freezing layers, 2017. URL https://arxiv.org/abs/1706.04983.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, Florence, Italy, 2019b. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://aclanthology.org/W19-4828.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mZn2Xyh9Ec.

Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Re. Flashattention: Fast and memory-efficient exact attention with IO-awareness. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=H4DqfPSibmx.

Yihe Dong, Lorenzo Noci, Mikhail Khodak, and Mufan Li. Is random attention sufficient for sequence modeling? disentangling trainable components in the transformer, 2025. URL https://arxiv.org/abs/2506.01115.

Aleksandr Dremov, David Grangier, Angelos Katharopoulos, and Awni Hannun. Compute-optimal quantization-aware training. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QpbtT95S95.

Goker Erdogan, Nikhil Parthasarathy, Catalin Ionescu, Drew A. Hudson, Alexander Lerchner, Andrew Zisserman, Mehdi S. M. Sajjadi, and Joao Carreira. Layerlock: Non-collapsing representation learning with progressive freezing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19461–19470, October 2025.

Hengyu Fu, Tianyu Guo, Yu Bai, and Song Mei. What can a single attention layer learn? a study through the random features lens. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 11912–11951. Curran Associates, Inc., 2023. doi: 10.52202/075280-0521. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/274db6bf1b01d8b4f07feaeb8c46f474-Paper-Conference.pdf.

Michael Hassid, Hao Peng, Daniel Rotem, Jungo Kasai, Ivan Montero, Noah A. Smith, and Roy Schwartz. How much does attention actually attend? questioning the importance of attention in pretrained transformers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1403–1416, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.101. URL https://aclanthology.org/2022.findings-emnlp.101/.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Huiqiang Jiang, YUCHENG LI, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=fPBACAbqSN.

Weixian Waylon Li, Yftah Ziser, Maximin Coavoux, and Shay B. Cohen. BERT is not the count: Learning to match mathematical statements with proofs. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3581–3593, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.260. URL https://aclanthology.org/2023.eacl-main.260/.

Weixian Waylon Li, Yuchen Niu, Yongxin Yang, Keshuang Li, Tiejun Ma, and Shay B Cohen. Spectral attention steering for prompt highlighting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XfLvGIFmAN.

Yuhan Liu, Saurabh Agarwal, and Shivaram Venkataraman. Autofreeze: Automatically freezing model blocks to accelerate fine-tuning, 2021. URL https://arxiv.org/abs/2102.01386.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20192–20204, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1035. URL https://aclanthology.org/2025.findings-acl.1035/.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 14014–14024, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/2c601ad9d2ff9bc8b282670cdd54f69f-Abstract.html.

Clement Neo, Shay B. Cohen, and Fazl Barez. Interpreting context look-ups in transformers: Investigating attention-MLP interactions. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16681–16697, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.930. URL https://aclanthology.org/2024.emnlp-main.930/.

Hikari Otsuka, Daiki Chijiwa, Yasuyuki Okoshi, Daichi Fujiki, Susumu Takeuchi, and Masato Motomura. The strong lottery ticket hypothesis for multi-head attention mechanisms, 2025. URL https://arxiv.org/abs/2511.04217.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. QuALITY: Question answering with long input texts, yes! In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5336–5358, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.391. URL https://aclanthology.org/2022.naacl-main.391/.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. Fixed encoder self-attention patterns in transformer-based machine translation. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 556–568, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.49. URL https://aclanthology.org/2020.findings-emnlp.49/.

Bowen Shen, Zheng Lin, Daren Zha, Wei Liu, Jian Luan, Bin Wang, and Weiping Wang. Pruning large language models to intra-module low-rank architecture with transitional activations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9781–9793, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.582. URL https://aclanthology.org/2024.findings-acl.582/.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10183–10192. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/tay21a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580.

Ziyan Wang, Enmao Diao, Qi Le, Pu Wang, Minwoo Lee, Shu-ping Yeh, Evgeny Stupachenko, Hao Feng, and Li Yang. From local to global: Revisiting structured pruning paradigms for large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35720–35739, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1653. URL https://aclanthology.org/2026.acl-long.1653/.

Tsung-Han Wu, Chun-Chen Hsieh, Yen-Hao Chen, Po-Han Chi, and Hung yi Lee. Input-independent attention weights are expressive enough: A study of attention in self-supervised audio transformers, 2020. URL https://arxiv.org/abs/2006.05174.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=EytBpUGB1Z.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=09iOdaeOzp.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, junxian guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. Duoattention: Efficient long-context LLM inference with retrieval and streaming heads. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cFu7ze7xUm.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture.

In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Huiyin Xue, Nafise Sadat Moosavi, and Nikolaos Aletras. Deconstructing attention: Investigating design principles for effective language modeling. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 708–727, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. ISBN 979-8-89176-298-5. doi: 10.18653/v1/2025.ijcnlp-long.40. URL https://aclanthology.org/2025.ijcnlp-long.40/.

Qingyue Yang, Jie Wang, Xing Li, Yinqi Bai, Tong Xialiang, Hui-Ling Zhen, Jianye HAO, Mingxuan Yuan, and Bin Li. Why attention patterns exist: A unifying temporal perspective analysis. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XhqoDBouWS.

Weiqiu You, Simeng Sun, and Mohit Iyyer. Hard-coded Gaussian attention for neural machine translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7689–7700, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.687. URL https://aclanthology.org/2020.acl-main.687/.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: transformers for longer sequences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Minjia Zhang and Yuxiong He. Accelerating training of transformer-based language models with progressive layer dropping. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14011–14023. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/a1140a3d0df1c81e24ae954d935e8926-Paper.pdf.

A Additional Method Details

We give the attention notation and calibration protocol, followed by the complete SAF algorithm and definitions of the alternative selectors and schedules.

A.1 Attention notation and calibration

Consider a model with L attention layers and H query heads per layer, giving LH heads in total. Let $\ell \in \{1, \dots, L\}$ index layers, $h \in \{1, \dots, H\}$ index heads, T denote sequence length, d denote the hidden dimension, and d_h denote the head dimension. For an input sequence, $\mathbf{R}_\ell \in \mathbb{R}^{T \times d}$ denotes the residual stream entering layer ℓ. The attention input is $\mathbf{X}_\ell = \mathrm{LN}_\ell(\mathbf{R}_\ell)$, where LN_ℓ is the pre-attention layer normalisation (Xiong et al., 2020). The projections are $\mathbf{Q}_{\ell h} = \mathbf{X}_\ell \mathbf{W}^Q_{\ell h}$, $\mathbf{K}_{\ell h} = \mathbf{X}_\ell \mathbf{W}^K_{\ell h}$, and $\mathbf{V}_{\ell h} = \mathbf{X}_\ell \mathbf{W}^V_{\ell h}$, with projection matrices in $\mathbb{R}^{d \times d_h}$ and outputs in $\mathbb{R}^{T \times d_h}$. Rotary position embeddings, when used, act on $\mathbf{Q}$ and $\mathbf{K}$ before the dot product (Su et al., 2024). Standard causal attention maps $\mathbf{X}_\ell$ to

$$\mathbf{S}_{\ell h}(\mathbf{X}_\ell) = \frac{\mathbf{Q}_{\ell h} \mathbf{K}_{\ell h}^\top}{\sqrt{d_h}}, \tag{5}$$

$$\mathbf{A}_{\ell h}(\mathbf{X}_\ell) = \mathrm{softmax}_{\mathrm{causal}}(\mathbf{S}_{\ell h}(\mathbf{X}_\ell)), \tag{6}$$

$$\mathbf{Z}_{\ell h}(\mathbf{X}_\ell) = \mathbf{A}_{\ell h}(\mathbf{X}_\ell) \mathbf{V}_{\ell h}. \tag{7}$$

Writing $\mathbf{W}^O_{\ell h} \in \mathbb{R}^{d_h \times d}$ for the output-projection block of head h, its contribution to the residual stream is $\mathbf{C}_{\ell h} = \mathbf{Z}_{\ell h} \mathbf{W}^O_{\ell h}$. Let $i, j \in \{1, \dots, T\}$ index query and key positions, respectively. The operator $\mathrm{softmax}_{\mathrm{causal}}$ masks future keys and normalises each query row, so $\mathbf{A}_{\ell h}(i, j) = 0$ for $j > i$, $\mathbf{A}_{\ell h}(i, j) \geq 0$, and $\sum_{j=1}^{i} \mathbf{A}_{\ell h}(i, j) = 1$. Therefore, the attention matrix is lower triangular (including the diagonal) and row-stochastic.

We measure attention at the replacement checkpoint in evaluation mode, keeping its weights fixed. The original 124M and 1B pretraining runs sample calibration sequences from the training corpus; the additional three-seed controls use a held-out calibration partition. Both protocols use calibration data separate from final reporting and leave the training-data sampler unchanged; all head scores and pattern targets are computed before replacement. Appendices C.1, C.3, and C.5 give the data partitions and calibration budgets.

A.2 Training procedure

Algorithm 1 specifies the one-time SAF recipe. The model first completes $t_\star$ of M optimiser updates, then measures attention on $\mathcal{D}_{\mathrm{cal}}$ using its current weights. Each $\mathcal{D}_t$ contains the micro-batches for one complete training update; LMUpdate applies the language-model objective, prescribed learning rate, and optimiser state ω. Calibration leaves model weights and the training-data order unchanged. The calibration partitions differ between the original pretraining runs and the supplementary controls, as specified in Appendices C.1, C.3, and C.5.

Selection, fitting, and installation. The algorithm considers candidates in increasing variance order, skips failed compact fits, and continues until k heads pass the fitting threshold or the candidates are exhausted. The requested rate is attained only if k fits pass. Variance determines candidate order, while $\varepsilon_{\mathrm{fit}}$ bounds the compact-fitting error. It represents batched fitting as a headwise loop; the model receives no training updates between candidate fits. Comparisons that prescribe identical head sets instead fit only the specified heads and reject a configuration if any fit fails, without substituting other heads. The additional fixed-token controls use this rule for variance-selected means (Appendix C.3). Fit initialises $\alpha(j) = 0$ and $\rho(\delta) = \log \max\{\eta_{\mathrm{rel}}(\delta)/(T - \delta), 10^{-9}\}$, then applies $J = 400$ Adam steps at learning rate $\gamma = 0.05$ to Equation 4, without early stopping. The acceptance threshold is $\varepsilon_{\mathrm{fit}} = 0.2$ nats. Computing $\widehat{\mathbf{P}}$ and its KL in the algorithm denotes evaluation of the fitted distribution; the FFT fitter obtains normalisers and KL from sufficient statistics without materialising that matrix. After installation, the fixed pattern parameters receive no language-model gradients; ordinary attention and the remaining model parameters, including value and output projections, continue to train.

Algorithm 1 Selective Attention Freezing (SAF) with one-time replacement.

Require: Model f_θ, optimiser state ω, training batches $\{\mathcal{D}_t\}_{t=1}^{M}$,
calibration sequences $\mathcal{D}_{\mathrm{cal}}$ of length T
Require: Replacement update step $t_\star$, rate r, fit steps J, fit learning rate γ, tolerance $\varepsilon_{\mathrm{fit}}$
Ensure: Trained f_θ, replaced heads $\mathcal{F}$, fixed pattern buffers $\mathcal{B}$

for $t = 1, \dots, t_\star$ **do**
  $(\theta, \omega) \leftarrow \mathrm{LMUPDATE}(\theta, \omega, \mathcal{D}_t, t)$
end for
Measure $\widehat{\mathbf{A}}_{\ell h}$ (Section 3.1) and $s^{\mathrm{var}}_{\ell h}$ (Equation 1) on $\mathcal{D}_{\mathrm{cal}}$ in evaluation mode, holding θ fixed
$\pi \leftarrow$ all head indices sorted by increasing $s^{\mathrm{var}}_{\ell h}$
$k \leftarrow \mathrm{round}(rLH)$; $\mathcal{F} \leftarrow \varnothing$; $\mathcal{B} \leftarrow \varnothing$
for head $u = (\ell, h)$ in π **do**
  if $|\mathcal{F}| = k$ **then**
    break
  end if
  Floor causal entries of $\widehat{\mathbf{A}}_u$ at 10^{-9}, then row-normalise to obtain $\mathbf{P}_u$
  Compute $\eta_{\mathrm{abs}}, \eta_{\mathrm{rel}}$ from $\mathbf{P}_u$ (Equation 4)
  $(\alpha_u, \rho_u) \leftarrow \mathrm{FIT}(\eta_{\mathrm{abs}}, \eta_{\mathrm{rel}}, J, \gamma)$
  Compute Z_u and $\widehat{\mathbf{P}}_u$ as in Equation 3
  $\kappa_u \leftarrow T^{-1} \sum_{i=1}^{T} \mathrm{KL}\Big(\mathbf{P}_u[i, 1{:}i] \,\|\, \widehat{\mathbf{P}}_u[i, 1{:}i]\Big)$
  if $\kappa_u \le \varepsilon_{\mathrm{fit}}$ **then**
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{u\}$; $\mathcal{B}[u] \leftarrow (\alpha_u, \rho_u, \log Z_u)$
  end if
end for
Verify that $|\mathcal{F}| = k$ for the requested replacement rate
Install $\mathcal{B}$ as fixed attention patterns for $\mathcal{F}$; retain value/output weights and optimiser state
Resume training mode with $\mathcal{F}$ and $\mathcal{B}$ fixed
for $t = t_\star + 1, \dots, M$ **do**
  $(\theta, \omega) \leftarrow \mathrm{LMUPDATE}(\theta, \omega, \mathcal{D}_t, t)$ using mixed-head attention
end for
return $f_\theta, \mathcal{F}, \mathcal{B}$

Schedule variants. The default uses one event near the training midpoint; earlier or later one-time replacement changes $t_\star$. For a schedule $\{(t_k, \mathcal{H}_k)\}_{k=1}^{K}$, the same calibration, fitting, and installation operations apply at each event to newly proposed heads $\mathcal{H}_k$. The gradual comparison fixes the ordered final head set and installs it in five groups, fitting each group's patterns at its replacement event. Adaptive controllers determine events from validation or training loss and restrict proposals to heads that still use ordinary attention. The validation-budget controller accepts an increment only if its paired loss change fits within the cumulative budget; the train-loss-triggered controller waits for a plateau and subsequent recovery, and can undo an unsuccessful increment. Previously installed patterns remain fixed while new candidates are measured. These alternatives modify the event schedule and acceptance policy, rather than making recalibration part of every training update; Appendix A.3 gives their detailed settings.

A.3 Alternative selector and schedule definitions

Section 3.1 defines the attention-variance and forward-KL scores. Appendix B.2 reports results for the gradient and output-aware scores defined below, including preliminary gradient comparisons, one-head diagnostics, and continuation experiments. For the gradient alternative, let t index optimiser updates, let $\mathcal{L}_t$ be the current minibatch language-model loss, and define

$$q^{(t)}_{\ell h} = \|\nabla_{\mathbf{W}^{Q}_{\ell h}} \mathcal{L}_t\|_F + \|\nabla_{\mathbf{W}^{K}_{\ell h}} \mathcal{L}_t\|_F, \qquad \overline{q}^{(t)} = (LH)^{-1} \sum_{\ell', h'} q^{(t)}_{\ell' h'}. \tag{8}$$

We remove the scale shared across heads at update t using

$$s^{\mathrm{grad}}_{\ell h}(t) = \beta s^{\mathrm{grad}}_{\ell h}(t-1) + (1-\beta) \frac{q^{(t)}_{\ell h}}{\overline{q}^{(t)}}, \qquad \beta = 0.9, \tag{9}$$

initialised by $s^{\mathrm{grad}}_{\ell h}(1) = q^{(1)}_{\ell h}/\overline{q}^{(1)}$. The symmetric rank combination is $s^{\mathrm{sum}}_{\ell h} = \mathrm{rank}(s^{\mathrm{var}}_{\ell h}) + \mathrm{rank}(s^{\mathrm{grad}}_{\ell h})$, with rank zero assigned to the smallest value across all LH heads. The asymmetric gradient-veto rule orders heads by $s^{\mathrm{var}}_{\ell h}$ but places heads with above-median $s^{\mathrm{grad}}_{\ell h}$ after the remaining candidates.

The output-aware scores use a separate set of N_f held-out sequences indexed by m and query positions $\mathcal{I}$ sampled evenly from the latter half of each sequence. The symbols $\mathbf{R}_{\ell,m}$ and $\mathbf{X}_{\ell,m}$ denote the residual input and normalised attention input from Appendix A.1 on sequence m. Define $\mathbf{V}^{(m)}_{\ell h} = \mathbf{X}_{\ell,m}\mathbf{W}^{V}_{\ell h}$, $\mathbf{A}^{(m)}_{\ell h} = \mathbf{A}_{\ell h}(\mathbf{X}_{\ell,m})$, and $\mathbf{C}^{(m)}_{\ell h} = \mathbf{A}^{(m)}_{\ell h}\mathbf{V}^{(m)}_{\ell h}\mathbf{W}^{O}_{\ell h}$. The vector $\mathbf{c}_{\ell h}$ concatenates $\mathbf{C}^{(m)}_{\ell h}[i]$ over m and $i \in \mathcal{I}$. The projected-output magnitude is $s^{\mathrm{mag}}_{\ell h} = \|\mathbf{c}_{\ell h}\|_2/\sqrt{\dim(\mathbf{c}_{\ell h})}$, and within-layer redundancy is

$$s^{\mathrm{red}}_{\ell h} = -\max_{h' \neq h} \frac{\langle \mathbf{c}_{\ell h}, \mathbf{c}_{\ell h'} \rangle}{\|\mathbf{c}_{\ell h}\|_2 \|\mathbf{c}_{\ell h'}\|_2}. \tag{10}$$

The maximum ranges over heads in the same layer, and the minus sign gives higher replacement priority to highly redundant heads.

Let $\widetilde{\mathbf{C}}^{(m)}_{\ell h} = \widehat{\mathbf{A}}_{\ell h}\mathbf{V}^{(m)}_{\ell h}\mathbf{W}^{O}_{\ell h}$, $\mathbf{O}^{(m)}_{\ell} = \sum_h \mathbf{C}^{(m)}_{\ell h}$, and $\mathbf{Y}^{(m)}_{\ell} = \mathbf{R}_{\ell,m} + \mathbf{O}^{(m)}_{\ell}$. Replacing only head h gives $\widetilde{\mathbf{Y}}^{(m)}_{\ell h} = \mathbf{Y}^{(m)}_{\ell} - \mathbf{C}^{(m)}_{\ell h} + \widetilde{\mathbf{C}}^{(m)}_{\ell h}$. The two replacement-influence scores are

$$s^{\cos}_{\ell h} = \mathbb{E}_{m,i}\left[1 - \cos\left(\mathbf{Y}^{(m)}_{\ell}[i], \widetilde{\mathbf{Y}}^{(m)}_{\ell h}[i]\right)\right], \tag{11}$$

$$s^{\mathrm{rel}}_{\ell h} = \mathbb{E}_{m,i}\left[\frac{\|\widetilde{\mathbf{C}}^{(m)}_{\ell h}[i] - \mathbf{C}^{(m)}_{\ell h}[i]\|_2}{\max\left(\|\mathbf{O}^{(m)}_{\ell}[i]\|_2, 10^{-8}\right)}\right]. \tag{12}$$

The local diagnostic compares these scores with the paired held-out NLL change caused by replacing one head with its fitted mean pattern.

For a schedule $\{(t_k, \mathcal{H}_k)\}_{k=1}^{K}$, heads $\mathcal{H}_k$ are replaced at update t_k. Let $\mathcal{F} = \bigcup_{k=1}^{K} \mathcal{H}_k$ be the final selected set and $\tau = t/M$ the intervention fraction of an M-update run. The one-time schedule replaces all heads in $\mathcal{F}$ at update t, while the gradual schedule partitions $\mathcal{F}$ into five nearly equal groups applied at evenly spaced updates. The validation-budget schedule accepts small groups in score order while their cumulative paired validation-perplexity cost remains below a specified relative budget. The train-loss-triggered schedule proposes a group when smoothed training loss satisfies a plateau test and waits for recovery before the next proposal. Its parameter z is the standard-deviation multiplier in the plateau, statistical-power, and recovery tests; larger z permits an earlier trigger. The adaptive schedules determine their final rates through their stopping rules and configured caps.

B Understanding the Selected Recipe

Our controlled study selects the post-softmax mean as the fixed pattern and attention variance as the head-selection score. Corpus-averaged attention has also been used as an input-independent replacement in prior probing work (Hassid et al., 2022). Both choices follow from the same objective for approximation by a fixed matrix: the mean gives the best fixed pattern, while the variance measures how accurately each head can be approximated by one.

B.1 Fixed-pattern reconstruction

Fix one layer and head, suppress the indices ℓ, h, and write $\mathbf{A}(\mathbf{X})$ for its attention matrix on input $\mathbf{X}$ and $\mathbf{P}$ for its input-independent replacement. Let $\|\cdot\|_F$ denote the Frobenius norm. For a fixed replacement pattern $\mathbf{P}$, define the expected reconstruction error per matrix entry as

$$\mathcal{R}(\mathbf{P}) = \frac{1}{T^2}\mathbb{E}_{\mathbf{X}}\left[\|\mathbf{A}(\mathbf{X}) - \mathbf{P}\|_F^2\right]. \tag{13}$$

Writing $\overline{\mathbf{A}} = \mathbb{E}_{\mathbf{X}}[\mathbf{A}(\mathbf{X})]$ for the population mean gives

$$\mathcal{R}(\mathbf{P}) = \mathcal{R}(\overline{\mathbf{A}}) + \frac{1}{T^2}\|\mathbf{P} - \overline{\mathbf{A}}\|_F^2. \tag{14}$$

Table 4: Fixed-pattern geometry: within-rate correlation between distance from the post-softmax mean and final perplexity increase.

Distance from mean	10%	25%	50%	75%	Stratified
Squared error	0.900	1.000	1.000	0.900	0.950
$D_{\mathrm{KL}}(\overline{\mathbf{A}}\|\mathbf{P})$	0.600	0.600	0.600	0.700	0.625
Squared Hellinger	0.800	0.900	0.900	1.000	0.900
Total variation	0.800	0.900	0.900	1.000	0.900

Because $\overline{\mathbf{A}}$ is an average of causal row-stochastic matrices, it is itself a valid attention matrix and minimises this reconstruction error. The minimum value $\mathcal{R}(\overline{\mathbf{A}})$ is the population attention variance per matrix entry, which $s^{\mathrm{var}}_{\ell h}$ in Equation 1 estimates from calibration inputs. The same objective therefore determines both what fixed pattern to use and which heads are easiest to replace.

This argument applies to attention reconstruction rather than directly to language-model loss. The empirical mean $\widehat{\mathbf{A}}_{\ell h}$ only estimates the population mean $\overline{\mathbf{A}}$, while language-model loss can also depend on value vectors, output projections, interactions between heads, and further training after replacement. This reconstruction analysis nevertheless shows that input-independent sampling has no inherent advantage under this objective: a sampled pattern introduces variation without conditioning that variation on the current input.

Probability geometry provides a complementary view of the two mean patterns. These centroid properties are instances of the broader connection between Bregman divergences and their corresponding mean representations (Banerjee et al., 2005). For one valid causal row, let $\mathbf{a}(\mathbf{X})$ be the random attention vector and $\mathbf{p}$ a fixed probability vector on the same support. The two directions of KL divergence select different fixed summaries:

$$\arg\min_{\mathbf{p}} \ \mathbb{E}_{\mathbf{X}}\left[D_{\mathrm{KL}}(\mathbf{a}(\mathbf{X}) \,\|\, \mathbf{p})\right] = \mathbb{E}_{\mathbf{X}}[\mathbf{a}(\mathbf{X})], \tag{15}$$

$$\arg\min_{\mathbf{p}} \ \mathbb{E}_{\mathbf{X}}\left[D_{\mathrm{KL}}(\mathbf{p} \,\|\, \mathbf{a}(\mathbf{X}))\right] = \frac{\exp(\mathbb{E}_{\mathbf{X}}[\log \mathbf{a}(\mathbf{X})])}{\sum_j \exp(\mathbb{E}_{\mathbf{X}}[\log a_j(\mathbf{X})])}. \tag{16}$$

The first identity follows from the cross-entropy term, and the second from a Lagrange multiplier enforcing $\sum_j p_j = 1$. If $\mathbf{a} = \mathrm{softmax}(\mathbf{s})$, the shared row-normalisation term cancels in the second expression, which becomes $\mathrm{softmax}(\mathbb{E}[\mathbf{s}])$. Thus, the post-softmax and sharp means are respectively the forward- and reverse-KL barycentres.

We analyse fixed-pattern distances to check whether these geometries are consistent with the observed ordering, without treating them as an explanation of language-model loss. For each fixed pattern, we compare its stored matrix $\mathbf{P}$ with the measured post-softmax mean on the same nested head subset. In these empirical comparisons, $\overline{\mathbf{A}}$ denotes this measured mean before compact fitting. Table 4 ranks the five patterns in Figure 3 separately within each replacement rate, so that the shared effect of replacement rate does not inflate the association with final perplexity. Squared distance has a stratified Spearman correlation of 0.950 with perplexity increase, compared with 0.625 for $D_{\mathrm{KL}}(\overline{\mathbf{A}}\|\mathbf{P})$ and 0.900 for both squared Hellinger distance and total variation. These correlations are descriptive because each within-rate comparison contains only five patterns and the post-softmax mean has zero squared distance by construction.

B.2 Selecting heads for continued pretraining

Head selection can be evaluated by the immediate cost of replacing one head or by the final loss after replacing a complete set and continuing training. These two evaluations need not agree because the latter also reflects interactions among selected heads and subsequent adaptation.

Preliminary 124M comparisons at 1K context use dense post-softmax means, 25% midpoint replacement, and one run per selector. Q/K gradient selection gives similar final perplexity to variance (26.558 versus 26.568), while the rank sum and gradient-veto rule increase it to 26.627 and 26.607, respectively; training with ordinary attention gives 26.382. These dense-pattern results are separate from the compact-pattern comparisons below.

Table 5: Immediate one-head diagnostics at the 4K RoPE midpoint. Spearman correlates scores with NLL changes; mean costs are relative perplexity increases.

Risk score	Spearman ↑	25% mean cost ↓	50% mean cost ↓
Attention variance, s^{var}	0.723	0.058%	**0.115%**
Residual cosine, $s^{\cos}$	**0.810**	**0.033%**	0.121%
Relative attention-output change, s^{rel}	0.703	0.052%	0.150%
Projected-output magnitude, s^{mag}	0.654	0.166%	0.129%
Projected-output redundancy, s^{red}	−0.013	0.320%	0.367%

Table 6: Final perplexity increase after joint head selection, midpoint replacement, and 2,500 continuation updates.

Context	Replaced heads	Variance ΔPPL	Forward KL ΔPPL	KL–variance (pp)
4K	25%	0.665%	1.068%	+0.402
4K	50%	2.649%	3.554%	+0.905
8K	25%	0.557%	0.836%	+0.279
8K	50%	2.191%	2.988%	+0.797
16K	25%	0.474%	0.876%	+0.402
16K	50%	1.802%	2.417%	+0.615

At the 4K midpoint, we test whether output-aware scores predict immediate replacement cost more accurately than attention variance. We fit a compact pattern to the measured post-softmax mean of each of 144 heads and measure the paired NLL change on held-out windows when replacing one head at a time. The score comparisons use the 118 heads that pass the compact-fitting acceptance threshold. Table 5 compares attention variance with four output-aware scores and reports their association with this local cost and the mean cost over their lowest-risk 25% and 50% subsets.

Residual cosine predicts the immediate one-head cost more closely and selects a 25% set with a smaller joint perturbation than variance. The sets share 23 of 36 heads, and their immediate PPL increases on 16 new common validation windows are 4.136% and 5.092%, respectively. After replacing both sets from the same update-2,500 checkpoint and continuing for the same 2,500 updates, however, their final PPL increases are 0.792% and 0.665%. Resampling the 598 matched 4K evaluation windows gives a residual-cosine-minus-variance PPL difference of 0.126% (95% interval: 0.102–0.150%). A small local perturbation from replacing one head therefore does not necessarily predict the final result after jointly replacing many heads and continuing training. The discrepancy may reflect interactions among selected heads, subsequent adaptation, or both.

We next compare the two input-dependence scores, attention variance and row-wise forward KL, after adaptation is complete. For every context length and replacement rate, both methods select a complete head set at the midpoint and undergo the same 2,500 continuation updates.

Attention variance gives the lower final perplexity increase in all six matched settings.

The selected heads are concentrated in earlier layers, and this distribution is consistent across seeds. In the three-seed 4K controls of Appendix C.3, the first four layers contain 21–24 of the 36 heads selected at 25% replacement (Figure 7a). Selection is also stable within each training run: the quarter-training and midpoint sets share 30–31 heads, compared with an expected 12.6–13.4 for independent selections with the same per-layer counts (Figure 7b). The recorded variance rankings have Spearman correlations of 0.959–0.962 between these stages. We compare layer distributions across seeds and head identities within a seed, since head indices need not represent the same function in independently trained models.

To characterise the selected patterns, we evaluate the ordinary-attention checkpoints on 128 common, non-overlapping 4K validation windows and 64 evenly spaced query positions in the latter half of each window. These inputs are separate from the final perplexity evaluation. For a causal query row $\mathbf{p} = (p_1, \ldots, p_i)$, we measure local mass $\sum_{j=\max(1,i-63)}^{i} p_j$, the largest probability $\max_j p_j$, first-token mass p_1, and normalised entropy $-\sum_{j=1}^{i} p_j \log p_j / \log i$. We average each quantity

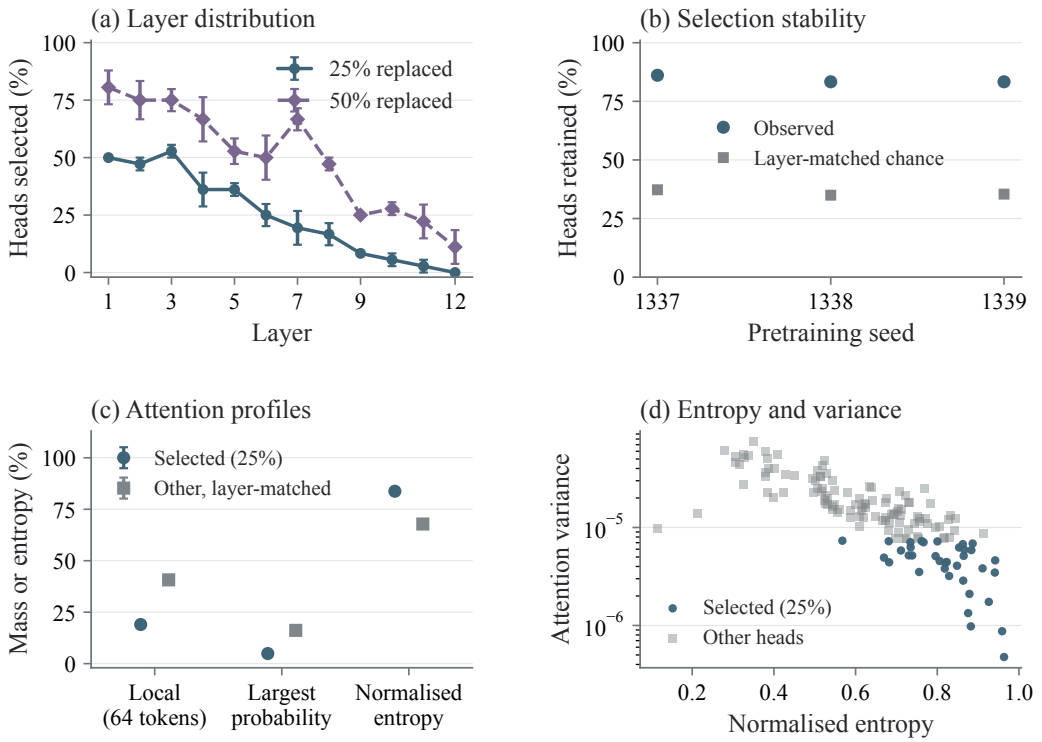

Figure 7: Selected-head diagnostics: (a) layer distribution; (b) quarter-to-midpoint retention; (c) midpoint attention profiles; (d) entropy and variance in the original one-head probe. Bars in (a,c) show mean ± SE over three seeds.

across inputs and query positions, then compare the 36 selected heads with unselected heads weighted to match their layer distribution. The selected heads spread attention more broadly: their mean normalised entropy is 0.837 versus 0.678, and their mass on the most recent 64 tokens, including the current token, is 19.0% versus 40.7% (Figure 7c). Only 0.023% of their attention falls on the first token on average, so first-token concentration does not characterise the selected set. These measurements describe attention shape rather than assigning linguistic functions or establishing attention sinks.

Variance remains associated with local replacement cost after adjusting for layer and entropy. We add entropy measurements to the original seed-1337 one-head diagnostic above, using its same checkpoint and 16 calibration windows, with the same 64 query positions as the profile analysis. Among its 118 eligible heads, variance and normalised entropy have Spearman correlation -0.798. The variance–replacement-cost correlation is 0.723 before adjustment, 0.733 after controlling for layer, and 0.543 after also controlling for entropy. For these partial rank correlations, we regress the ranked variance and ranked one-head NLL change on layer indicators, additionally include ranked entropy in the latter comparison, and correlate the residuals. Figure 7d shows all 144 heads, highlighting the eligible 25% selection. Thus, the score favours diffuse attention but retains information about local replacement cost beyond layer and entropy; the continuation and pruning comparisons test whether retaining that attention as a fixed pattern benefits the final model.

A matched-input comparison with the final ordinary-attention checkpoints shows that rankings continue to develop after quarter-training. Using the same 128 windows and sampled query rows at every stage, variance ranks correlate with their final ordering at 0.869–0.904 at quarter-training and 0.945–0.966 at the midpoint. These later-query diagnostics complement the recorded full-matrix selection scores: substantial overlap is already present early, while the midpoint ordering is more representative of the final model.

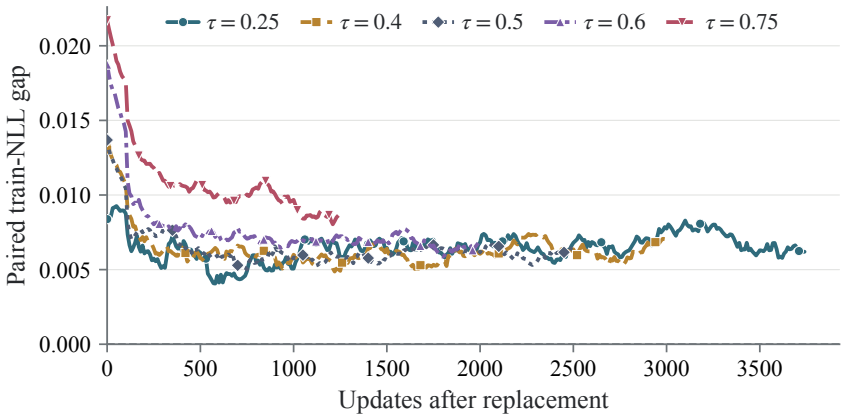

Figure 8: Paired train-loss gap after replacing 25% of heads at training fraction τ.

B.3 Adaptation after replacement

Replacement time introduces a second factor beyond the static reconstruction objective. Earlier replacement leaves more updates for the model to adjust, while later replacement uses attention statistics from a more developed model. In the replacement-time comparison, the immediate replacement cost increases from 2.72% at $\tau = 0.25$ to 7.99% at $\tau = 1$. The observed training trajectories directly support this adaptation effect. Figure 8 subtracts the ordinary-attention run's minibatch loss from each replacement run at the same update and data order and applies a centred 210-update moving average. Two hundred updates after replacement, the smoothed train-NLL gap is 0.0061 nats at $\tau = 0.25$ and 0.0123 nats at $\tau = 0.75$; by the final update, the corresponding gaps are 0.0062 and 0.0086 nats. Continued training can therefore offset part of the replacement cost, while late replacement leaves less time for recovery. The head diagnostics above show substantial selection overlap between quarter-training and the midpoint, alongside further changes towards the final ranking. The timing comparison therefore concerns both the fixed approximation available at replacement and the subsequent adaptation, rather than simply whether the selected heads have stabilised.

C Additional Experimental Details

We first give the controlled pretraining configurations and pattern diagnostics, then compare replacement with pruning under matched budgets. The remaining subsections cover 124M finetuning, 1B training and evaluation, and replacement in pretrained Qwen3-4B.

C.1 Controlled pretraining results

We train a 124M-parameter decoder with 12 Transformer layers, 12 heads per layer, hidden dimension $d = 768$, and context length $T = 4096$ on FineWeb-Edu (Penedo et al., 2024). We compare learned absolute position embeddings with RoPE (Su et al., 2024). Each run performs 5,000 optimiser updates with 491,520 target tokens per update, for a total of 2.4576B target tokens. This budget is approximately 20 tokens per parameter (Hoffmann et al., 2022). The 4K quality runs use micro-batch size $B = 2$ and 60 gradient-accumulation steps. The 8K and 16K quality runs use micro-batch size one with 60 and 30 accumulation steps, respectively, preserving the tokens per update and total training budget. The method-selection comparisons use seed 1337. After selecting two operating points from the replacement-rate comparison, we repeat these configurations and their ordinary-attention baselines with seeds 1338 and 1339.

Replacement runs start from the same pretrained checkpoint, retaining the remaining training configuration and data order. Pattern comparisons use the same selected heads, while rate comparisons use nested subsets of the same head ranking. Unless replacement timing is being studied, heads are replaced halfway through training. Every replacement run records the source checkpoint hash, parent checkpoint hash, training configuration, selected head list, fitted pattern, evaluation interval, and final

checkpoint hash. Replacement-time comparisons recompute the ordering at each parent checkpoint and use that ordering to pair mean and random patterns.

Table 7 summarises the controlled alternatives defined in Section 3. We treat position encoding as a model control rather than a choice introduced by attention replacement.

Table 7: Method choices evaluated in controlled 124M pretraining.

Method choice	Alternatives evaluated
Fixed pattern	Post-softmax mean; sharp mean; Gaussian sample; Dirichlet sample; structured random
Head selection	Attention variance; row-wise forward KL; Q/K gradient norm; rank combinations; projected-output magnitude and redundancy; attention-residual influence
Amount and timing	Rates: 10%, 25%, 50%, 75%; times: 25%, 40%, 50%, 60%, 75%, 100%; one-time, gradual, validation-budget, and train-loss-triggered schedules
Representation and execution	Exact dense or absolute-plus-relative pattern; reference or fused mixed-head execution

Gaussian, Dirichlet, and sharp patterns use dense storage in the pattern-content comparison; the mean is tested in both dense and absolute-plus-relative forms, with the latter used for systems measurements. Dirichlet concentration is scaled by 16, and structured-random patterns are reused across matched rates and replacement times. Gradual replacement uses $K = 5$ groups with the same final heads $\mathcal{F}$ and rate as one-time replacement; we also compare adaptive schedules at matched rates. Controller settings are given in Appendix A.3.

We additionally compare the selected recipe with head pruning and random head selection over three seeds at the same token budget. Pruning uses the variance-selected head sets; random selection retains fitted means and matches the number of replaced heads in each layer. These runs use $B = 8$, 15 accumulation steps, and a shuffled non-overlapping data order, with their own ordinary-attention references (Appendix C.3).

Optimisation and calibration. We use AdamW with a peak learning rate of 6×10^{-4}, linear warm-up over 500 updates, and cosine decay to 6×10^{-5} at update 5,000. The optimiser uses $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.1, and gradient clipping at norm 1; dropout is zero. Calibration uses 16 micro-batches at the training context length: 32 sequences at 4K and 16 at 8K and 16K, sampled from the configured training-data interval with a separate random-number generator. These forward passes collect attention statistics without optimisation and do not consume the sequence of training batches. The controller partition of the validation data is used for loss-based acceptance checks and before/after intervention probes; final perplexity uses a separate reporting partition.

Quality and systems measurements. The 4K systems sweep holds tokens per update fixed while using micro-batches two, four, and eight with 60, 30, and 15 accumulation steps. The cross-context systems comparison fixes the number of tokens per micro-batch at 32,768 with $(T, B) = (4096, 8), (8192, 4), (16384, 2)$ and 15 accumulation steps. Pretraining speedups are medians of within-pair ratios of ordinary-attention to replaced-model median update times; the displayed times are medians across pairs. These measurements include gradient accumulation, backward computation, clipping, the AdamW update, and gradient reset, but exclude data loading and one-time intervention costs. Some quality runs used the earlier dense fitter, while subsequent runs used an FFT fitter with separately checked fit agreement. Their intervention times cannot be combined with the update benchmark to establish total training speedup. Ordinary-attention heads use Flash scaled dot-product attention; every replacement run verifies its head mask, fused dispatch, and zero query/key gradients for replaced heads. We evaluate on 2,449,408 target tokens from a held-out interval not used to fit the patterns and store per-token log probabilities for paired comparisons. Using the notation from Section 4, we report

$$\Delta\mathrm{PPL}(\%) = 100 \left(\frac{p_{\mathrm{rep}}}{p_{\mathrm{ord}}} - 1 \right). \tag{17}$$

Table 8 compares six one-time replacement stages. Table 27 reports the dense and compact pattern representations under both position encodings. Figure 3 gives the fixed-pattern comparison with numerical values in each cell.

Table 8: Timing at 25% replacement with the post-softmax mean. τ: fraction of training completed; $M - t$: remaining updates; ΔPPL: final perplexity increase.

τ	$M - t$	ΔPPL (%)
25%	3750	0.724
40%	3000	0.679
50%	2500	**0.662**
60%	2000	0.711
75%	1250	0.936
100%	0	8.168

Table 9: Single-seed comparison of preset and adaptive replacement schedules.

Comparison	Setting	Rate	ΔPPL (%)
Preset schedule	one replacement	25.00%	0.662
	gradual, five groups	25.00%	0.689
	one replacement	50.00%	2.649
	gradual, five groups	50.00%	2.832
Adaptive schedule	validation budget 1%	6.25%	0.093
	validation budget 2%	12.50%	0.348
	train-loss triggered, $z = 3.0$	14.58%	0.409
	train-loss triggered, $z = 1.5$	50.00%	2.119

Table 10 repeats the one-time and train-loss-triggered schedules with three seeds at matched final rates. Values are reported as mean $\pm$ standard error, and the time ratio is train-loss-triggered time divided by one-time-schedule time.

Table 10: Three-seed comparison of one-time and train-loss-triggered schedules at matched final rates.

Rate	One-time ΔPPL (%)	Train-loss ΔPPL (%)	Difference (pp)	Time ratio (triggered/one-time)
25%	0.768 ± 0.067	0.858 ± 0.180	$+0.091 \pm 0.123$	1.027×
50%	2.492 ± 0.083	2.495 ± 0.189	$+0.003 \pm 0.267$	1.066×

Table 11 reports the corresponding post-replacement update speed and peak allocation at the fixed 4K systems configuration.

Table 11: Post-replacement training speed and memory at $T = 4096$ and batch size eight.

Replaced heads	Update speedup	Peak allocated	Saving
0%	1.000×	32.21 GiB	0.00 GiB
10%	1.016×	31.85 GiB	0.36 GiB
25%	1.056×	31.58 GiB	0.63 GiB
50%	1.119×	31.24 GiB	0.98 GiB
75%	1.215×	30.84 GiB	1.38 GiB

C.2 Gaussian and Dirichlet distribution diagnostics

The Gaussian and Dirichlet candidates model variation in different spaces. The Gaussian candidate fits a normal marginal to each pre-softmax entry across inputs, then applies softmax to obtain a valid probability row. The Dirichlet candidate instead models the complete post-softmax row jointly: its support already enforces nonnegative entries and a unit row sum. It therefore needs neither a

second softmax nor renormalisation. In both cases we model variation across calibration inputs, not uncertainty conditioned on the next input.

For the Dirichlet fit, suppress the head indices (ℓ, h) and write $\mathbf{m}_i = \widehat{\mathbf{A}}[i, 1{:}i]$ for the empirical mean of causal row i. Let $u_{ij} = (N-1)^{-1} \sum_{n=1}^{N} (\mathbf{A}^{(n)}(i,j) - \widehat{\mathbf{A}}(i,j))^2$ be the sample variance of entry (i, j) across the N calibration inputs. The row model and its moment estimate are

$$\mathbf{a}_i \sim \mathrm{Dirichlet}(c_i \mathbf{m}_i), \qquad c_i^{\mathrm{mom}} = \frac{1 - \|\mathbf{m}_i\|_2^2}{\sum_{j=1}^{i} u_{ij}} - 1, \tag{18}$$

where $\mathbf{a}_i$ is one sampled row and c_i is its total concentration. The estimate follows by summing $\mathrm{Var}[a_{ij}] = m_{ij}(1 - m_{ij})/(c_i + 1)$ over keys, with m_{ij} denoting entry j of $\mathbf{m}_i$. We multiply the estimated concentration by a chosen scale and bound it to $[10^{-2}, 10^{6}]$; nearly deterministic rows use the mean. For numerical sampling, Dirichlet parameters are floored at 10^{-6}. A larger concentration produces rows closer to the mean, but using a single concentration also constrains the covariance between keys. In particular, for distinct keys $j, k \leq i$, the row model imposes $\mathrm{Cov}[a_{ij}, a_{ik}] = -m_{ij} m_{ik}/(c_i + 1)$, rather than fitting their covariance independently. These constraints motivate our evaluation of the fit: valid probability vectors need not reproduce the observed attention distribution.

Our preliminary diagnostics use a 124M checkpoint at update 2,500 with $T = 1024$, separate from the main 4K comparison. We process 500 non-overlapping 1,024-token sequences from the FineWeb-Edu validation split with the model; these are text excerpts not used for pretraining, not model-generated continuations. For 36 selected heads, we record pre-softmax scores and reconstruct the corresponding probability rows at query positions $i \in \{64, 128, 256, 512, 768, 1024\}$. Thus, a fixed head and a fixed pair (i, j) yield 500 scalar observations across different inputs; a fixed head and query yield 500 entire probability vectors. We fit distributions on 300 sequences and reserve the remaining 200 for visual checks. Head labels in the diagnostic figures retain the zero-based layer and head indices used in the implementation.

Figure 9 compares held-out cumulative distributions before and after softmax, using the four heads shown in the original diagnostic. Each empirical curve uses the 200 held-out sequences; the Gaussian logit curve is analytic, while the probability curves use 500 generated rows per family. The Gaussian approximates some logit marginals closely, but its fit varies across heads, and sampling logits independently need not preserve the resulting attention distribution. The Dirichlet samples also differ visibly from the empirical probabilities despite obeying the row-sum constraint. For a quantitative check beyond individual entries, we compare distributions of complete square-root-transformed rows using energy distance. Distances are divided by the upper quartile of distances between two empirical subsets over 16 repeated splits, with matched subset sizes of 150 and at most 96 rows per energy estimate. Across the 36 heads and six query positions, median normalised distances are 1.10 for whole-row resampling from the calibration set, 1.44 for Gaussian sampling, and 2.58 for Dirichlet sampling at concentration scale 4, each averaged over four generated sets before aggregation. Whole-row resampling provides a reference for finite-sample variability; neither parametric family reproduces every head's distribution. The concentration scale 4 used in these diagnostics differs from scale 16 in the main pretraining comparison. The quality of the density fit is distinct from language-model quality when a single sampled pattern is retained.

Figure 10 complements the entry-wise check with a lower-dimensional view of complete attention profiles from the same checkpoint. This earlier diagnostic uses a separate collection of 128 non-overlapping validation sequences and queries in the later half of each sequence. For each query row, we sum attention mass into 12 disjoint query–key distance bins: self, previous token, distances 2–3, 4–7, and successive powers-of-two ranges, with the last bin collecting overflow. Each input therefore gives one 12-dimensional probability vector at each measured query position. We fit a separate Dirichlet to these binned vectors for each query position and use concentration scale 4. This is a fit in the binned space, not a fit to individual matrix entries. For each head, we pool positions and plot 512 empirical and 512 generated profiles. We apply an elementwise square root followed by coordinate-wise standardisation fitted on the empirical profiles; we then fit UMAP on these empirical profiles and use it to project the generated profiles and the mean. The square root makes the unstandardised distances proportional to Hellinger distance, but standardisation reweights the coordinates and the UMAP plot is not a quantitative distance test. Differences in coverage reveal

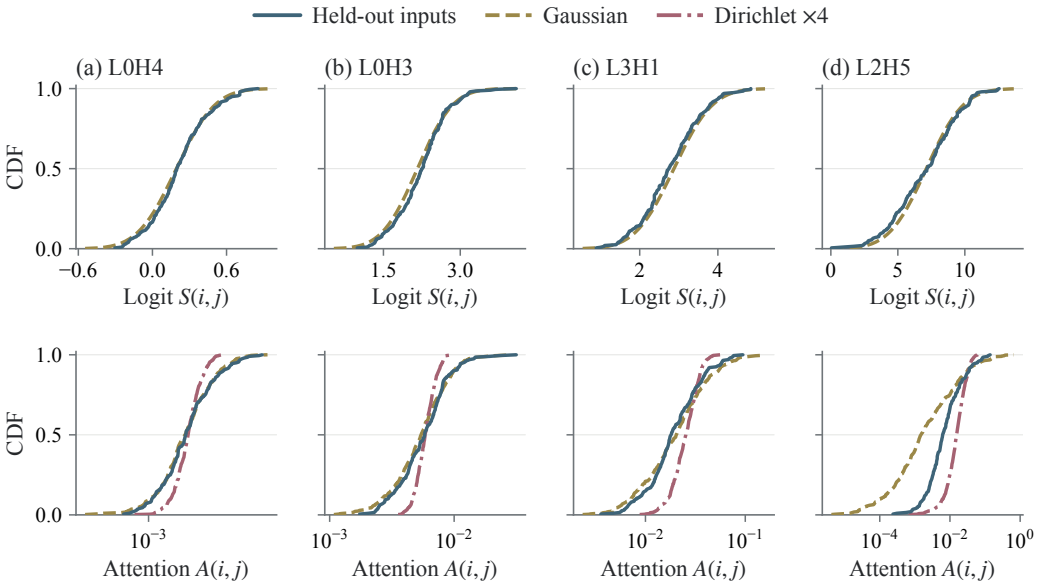

Figure 9: Fixed-entry CDFs at $i = 1024$, $j = 1023$: logits (top) and attention probabilities (bottom, logarithmic horizontal axis).

where the fitted samples miss empirical structure; visible clusters alone do not establish a mixture model or identify a uniquely suitable distribution.

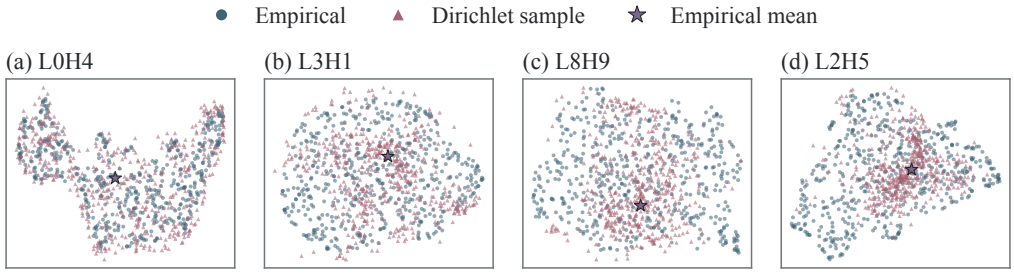

Figure 10: UMAP of empirical binned attention profiles and fitted Dirichlet samples at concentration scale 4.

C.3 Replacement and pruning controls

We compare replacement and pruning first on the same head sets, then with a selector designed for pruning. The matched-token, matched-time, and later-intervention studies use three pretraining seeds; training the pruned architecture from initialisation uses one. We then evaluate the resulting checkpoints on downstream tasks and measure the computation retained by fixed-pattern token mixing.

C.3.1 Matched-token pretraining

We compare SAF with head pruning and random head selection in 124M RoPE models at 4K context. For each of seeds 1337–1339, we report an ordinary-attention reference and seven continuations, giving 24 complete training paths. All paths perform 5,000 updates and process 2.4576B target tokens, using micro-batch size eight and 15 accumulation steps on one GH200. The architecture and AdamW settings follow Appendix C.1: 500 linear warm-up updates to 6×10^{-4}, followed by cosine decay to 6×10^{-5} at update 5,000. These comparisons use shuffled non-overlapping

training windows and separate ordinary-attention references from the earlier $B = 2$ quality runs. Each continuation restores the exact parent weights, AdamW state, random-number-generator state, and sampler position; subsequent training data are matched within each seed. Midpoint replacement occurs at update 2,500; the earlier 25% setting uses update 1,250.

Calibration uses 32 held-out sequences of length 4,096 in 16 batches of two, disjoint from the reporting interval. The mean uses 400 Adam fitting steps at learning rate 0.05, without early stopping; these matched-head controls abort if any selected fit exceeds 0.2 nats, rather than substituting another head. Across the twelve midpoint mean runs (three seeds, two rates, and two budgets), the selected sets equal the raw variance top-k sets: no heads are rejected or substituted, and the largest selected-head fitting KL is 0.159 nats. Thus, the fitting constraint does not change which heads are compared in these controls; this check concerns the selected sets, not all 144 candidates. Pruning uses exactly the same heads as the variance-selected mean at each rate. Pruning removes the selected heads' contributions, their query/key/value projection rows, and the corresponding output-projection columns, preserving the optimiser state of retained parameters. Random selection draws heads within each layer, matching the variance-selected counts and applying the same fitting threshold; it retains fitted mean patterns rather than using random patterns. The earlier intervention recalibrates and selects heads at its own checkpoint.

All final perplexities use the same 2,449,408 reporting tokens, with attention-projection caches refreshed before evaluation. Relative changes are computed against each seed's ordinary-attention reference before averaging; standard errors use the sample standard deviation divided by $\sqrt{3}$. Table 12 combines absolute results with paired differences from midpoint mean replacement at the same rate. Relative perplexity changes appear in Table 1; the additional early 25% setting gives $0.772 \pm 0.021\%$. Total training time includes training with ordinary attention before replacement and continuation afterwards, including calibration, head scoring, fitting, compilation, periodic monitoring, and intermediate checkpointing. It excludes queueing, final reporting evaluation, and checkpoint export. Configurations run in separate GH200 allocations; these full-path times differ from the controlled paired update benchmarks in Table 2. The matched-time comparison in Appendix C.3.3 uses an elapsed-time learning-rate schedule instead.

Table 12: Matched-token controls: mean ± SE over three pretraining seeds. Paired columns subtract midpoint mean + variance at the same rate.

				Paired differences	
Setting (rate)	PPL ↓	Training (min)	Intervention (s)	ΔPPL (pp)	Time (s)
Ordinary attention	23.896 ± 0.038	188.7 ± 2.7	—	—	—
Mean + variance (25%)	24.054 ± 0.041	184.7 ± 1.6	54.9 ± 6.0	—	—
Mean + variance (50%)	24.492 ± 0.035	178.7 ± 2.0	46.3 ± 1.2	—	—
Pruning + variance (25%)	24.088 ± 0.042	178.9 ± 1.6	46.3 ± 3.1	$+0.1428 \pm 0.0040$	-343.9 ± 27.8
Pruning + variance (50%)	24.621 ± 0.049	168.3 ± 1.3	50.8 ± 5.8	$+0.5378 \pm 0.0622$	-620.9 ± 65.3
Mean + random (25%)	24.202 ± 0.033	183.4 ± 1.2	53.7 ± 5.6	$+0.6186 \pm 0.0743$	-74.2 ± 23.7
Mean + random (50%)	24.763 ± 0.026	178.8 ± 1.8	57.5 ± 2.8	$+1.1344 \pm 0.0695$	$+9.1 \pm 63.6$
Mean + variance (25%, early)	24.080 ± 0.037	183.0 ± 1.1	49.2 ± 4.1	$+0.1085 \pm 0.0170$	-99.7 ± 31.0

One-time intervention cost. Table 13 separates attention measurement from fitting the compact parameters. For the 124M controls, total intervention time includes calibration, ranking, fitting, the KL check, and installation. Fitting and installation together take 3.30 ± 0.17 s at 25% replacement and 5.40 ± 0.10 s at 50%; ranking takes less than 0.012 s in every run. The two rates have separate calibration measurements, which account for the variation in total intervention time. For the 1B runs, the total is measured on rank zero through synchronisation and broadcast of the fitted state; ranking takes 0.019 and 0.021 s at 25% and 50% replacement, respectively. Separate before/after diagnostic evaluations add 3.68 and 2.76 s at 1B and are excluded from this table. The 124M entries apply to these $B = 8$ controls using the FFT fitter; the earlier $B = 2$ quality experiments include both dense and FFT fitting implementations.

Table 13: One-time calibration and fitting costs in seconds. The 124M controls report mean ± SE over three seeds; the 1B runs use one seed. Fitting is included in the total.

Model	Rate	Capture and mean	Fit α, ρ	Total intervention
124M, 4K	25%	51.56 ± 5.96	3.20 ± 0.16	54.86 ± 6.04
124M, 4K	50%	40.89 ± 1.11	5.24 ± 0.09	46.30 ± 1.22
1B, 8K	25%	1422.37	39.56	1464.41
1B, 8K	50%	1313.35	76.28	1392.64

C.3.2 Pattern content and trainability

We compare frozen and trainable compact means at 25% replacement in 124M RoPE models at 4K context, with $B = 8$ and 15 accumulation steps. Across three pretraining seeds, both variants restore the same 2,500-update ordinary-attention checkpoint, AdamW state, and training-data order, then continue to 5,000 updates and 2.4576B tokens. They use the same 36 variance-selected heads and exactly the same fitted initial parameters, with every selected fit below the 0.2-nat threshold. Only the trainable variant adds α, ρ to AdamW, using the model learning rate and betas, zero weight decay, and initially zero optimiser moments for these new parameters. Both variants use the fused mixed-head forward and value-gradient implementation; the trainable variant additionally computes pattern gradients and refreshes the normalisers after each update.

Table 14a reports fresh frozen and trainable runs in separate GH200 allocations. Update time averages the last 100 complete optimiser updates; continuation time includes loading, intervention, compilation, and monitoring, with final reporting and checkpoint export excluded. Perplexity uses the same reporting tokens as Table 12. The paired relative PPL change from learning the patterns is $-0.010 \pm 0.006\%$, while update and continuation times increase by $11.9 \pm 1.2\%$ and $12.4 \pm 1.5\%$, respectively.

For MQAR, we use the ordinary-attention and frozen-mean checkpoints from Table 12, together with the trainable endpoints from Table 14a. All three groups use matched 2.4576B-token pretraining budgets, with midpoint replacement where applicable. The task protocol follows Appendix C.3.6, with task seed 1337 for each pretraining seed: 1,500 updates at 512 tokens and eight pairs. Trainable patterns continue learning during task adaptation with zero pattern weight decay; the remaining model parameters use weight decay 0.01. Training data and all 512-example test sets are shared across methods and pretraining seeds. This comparison evaluates pattern choices at the midpoint; Figure 5 evaluates pruning after later, matched-time pretraining.

In Table 14b, continued pattern learning gives slightly higher mean accuracy than frozen mean in the length-only tests, whereas frozen mean gives higher means throughout the fixed-length pair-count sweep.

Table 14: Pattern controls at 25% replacement: mean ± SE over three pretraining seeds.

(a) Matched-token pretraining: 124M, 4K, 2.4576B tokens

Pattern	PPL ↓	Update (s) ↓	Continuation (min) ↓	Peak (GiB) ↓
Frozen mean	24.055 ± 0.040	2.159 ± 0.012	90.96 ± 0.56	31.668 ± 0.002
Trainable mean	24.052 ± 0.042	2.416 ± 0.013	102.27 ± 0.82	31.801 ± 0.002

(b) MQAR accuracy (%) ↑: midpoint, matched-token checkpoints

Tokens	Pairs	Ordinary attn.	Frozen mean	Trainable mean
512	8	99.58 ± 0.15	99.60 ± 0.06	99.54 ± 0.07
512	16	71.78 ± 2.47	81.36 ± 1.58	78.44 ± 3.69
512	24	52.71 ± 2.06	60.89 ± 1.61	59.67 ± 5.69
512	32	42.13 ± 1.49	49.08 ± 3.04	47.77 ± 5.62
512	48	30.00 ± 1.30	35.39 ± 3.74	34.67 ± 5.00
512	64	23.53 ± 1.34	28.30 ± 4.18	27.04 ± 4.19
2,048	8	97.01 ± 0.46	97.53 ± 0.44	97.99 ± 0.73
4,096	8	88.76 ± 0.88	88.89 ± 0.93	89.65 ± 1.42

C.3.3 MATCHED-TIME PRETRAINING

To test whether cheaper updates improve quality within a fixed training time, we restore each seed's common 2,500-update checkpoint. The remaining allowance is the ordinary-attention reference's total training time minus the time spent reaching the shared checkpoint. Every continuation receives this remaining allowance, including loading, compilation, calibration, selection, fitting, installation, and monitoring. Queueing and final reporting or checkpoint export are excluded. The learning-rate schedule advances from its midpoint value to 6×10^{-5} as a function of elapsed time, identically for ordinary attention, mean, pruned, and random-head models. Training follows the same data order within each seed, but faster models process more tokens. All final perplexities use the 2,449,408-token reporting set from the matched-token comparison.

Table 15 gives absolute perplexities and update counts for the comparison in Table 1. The full-path budgets span 187.3–188.1 minutes across seeds; differences of a few seconds within a seed arise from stopping before the next update would exceed the allowance. Runs use separate GH200 allocations; the paired kernel benchmarks instead alternate models on the same GPU. Random selection preserves the variance-selected number of heads in each layer and uses fitted mean patterns under the same time allowance. Its perplexity penalty exceeds variance selection by 0.786 ± 0.082 and 0.866 ± 0.154 percentage points at 25% and 50%, respectively, with the same ordering in all three seeds.

Table 15: Matched-time controls: mean $\pm$ SE over three pretraining seeds. Updates and times include the initial training with ordinary attention.

Operation + selection	Rate	PPL ↓	Updates	Time (min)
Ordinary attention	0%	23.882 ± 0.023	5008 ± 13	187.65 ± 0.25
Mean + variance	25%	23.991 ± 0.006	5109 ± 11	187.64 ± 0.25
Mean + variance	50%	24.338 ± 0.020	5221 ± 18	187.64 ± 0.25
Pruning + variance	25%	23.909 ± 0.024	5267 ± 30	187.64 ± 0.25
Pruning + variance	50%	24.115 ± 0.003	5685 ± 36	187.65 ± 0.25
Mean + random heads	25%	24.178 ± 0.024	5097 ± 28	187.64 ± 0.25
Mean + random heads	50%	24.545 ± 0.048	5242 ± 11	187.65 ± 0.26

C.3.4 LATER REPLACEMENT AND PRUNING-SPECIFIC SELECTION

We compare 25% replacement and pruning across three pretraining seeds after 5,000 updates with ordinary attention, or 2.4576B tokens. Within each seed, all continuations restore the same checkpoint and optimiser state, retaining 4K context, micro-batch eight, and 15 accumulation steps. The token-matched group adds 2,500 updates, giving 3.6864B tokens in total. Its continuation learning rate decays from 6×10^{-5} to 6×10^{-6}. The time-matched group instead receives its seed's measured token-matched continuation time with ordinary attention, 5,648–5,670 seconds, and uses the same endpoint learning rates with elapsed-time cosine decay. The allowance includes selection and installation; continuations stop before another update would exceed it. Final reporting uses 4,997,120 held-out tokens, so absolute perplexities should be compared within this study rather than with Table 15.

The pruning-specific criterion measures loss sensitivity through head-output gates (Michel et al., 2019). For calibration sequence n, multiply head ℓh's output by a scalar gate $\xi_{\ell h}$ before the output projection, and let $\mathcal{L}^{(n)}(\boldsymbol{\xi})$ be its mean-token negative log-likelihood with gate collection $\boldsymbol{\xi}$. Ordinary attention has all gates equal to one. The gate-Taylor importance and normalised pruning score are

$$I_{\ell h} = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{\partial \mathcal{L}^{(n)}(\boldsymbol{\xi})}{\partial \xi_{\ell h}}\bigg|_{\boldsymbol{\xi}=\mathbf{1}}\right|, \qquad s_{\ell h}^{\mathrm{gate}} = \frac{I_{\ell h}}{\sqrt{\sum_{h'=1}^{H} I_{\ell h'}^2}}. \tag{19}$$

Zero-norm layers receive zero scores. We take the absolute derivative before averaging across sequences and remove the 36 heads with the smallest normalised scores globally, breaking ties by layer and head index. This is a one-time use of the importance criterion, not the iterative pruning procedure of Michel et al. (2019). Calibration uses exactly the same 32 held-out sequences as variance selection. In the matched-time group, the selected set overlaps with the variance-selected set in 19, 19, and 21 of 36 heads across seeds; per-layer removal counts can also differ. Both pruning variants remove the corresponding projection rows and columns and preserve the retained AdamW state.

SAF and variance pruning retain exactly the same selected heads across token and time budgets. Independent gate-Taylor scoring selects 36, 35, and 35 common heads across budgets: two seeds exchange the nearly tied 36th and 37th candidates. The gate-Taylor budget comparison therefore also includes this small selection difference.

Table 16 combines pretraining quality with finetuning of the matched-time checkpoints. The paired finetuning differences, SAF minus gate-Taylor, are $+0.765 \pm 0.894$ points on SST-2 and -0.051 ± 0.795 on BoolQ, with reversals across pretraining seeds.

Training the reference-derived, variance-pruned architecture from initialisation for 7,500 updates gives PPL 22.683 in seed 1337, compared with 23.049 when introducing the same pruning after 5,000 updates. Its 248.26-minute training time excludes the cost of obtaining the reference-derived layout; the corresponding full-path times for ordinary attention, SAF, variance pruning, and gate-Taylor pruning are 286.86, 282.87, 275.43, and 275.93 minutes in this seed.

Table 16: Later 25% intervention: mean ± SE across three pretraining seeds. Finetuning uses the matched-time checkpoints and one fixed seed.

	Pretraining PPL ↓		Finetuning accuracy (%) ↑	
Operation + selection	Matched tokens	Matched time	SST-2	BoolQ
Ordinary attention	22.563 ± 0.010	22.561 ± 0.011	87.00 ± 0.33	71.78 ± 0.46
SAF + variance	22.923 ± 0.013	22.895 ± 0.012	87.73 ± 0.46	71.27 ± 0.67
Pruning + variance	23.019 ± 0.028	22.953 ± 0.026	87.81 ± 0.34	71.75 ± 0.29
Pruning + gate-Taylor	22.865 ± 0.019	22.798 ± 0.021	86.96 ± 0.49	71.33 ± 0.16

C.3.5 Downstream finetuning

Tables 16 and 17 distinguish pretraining variation from finetuning variation. The former and the midpoint group in the latter vary pretraining seeds while fixing the finetuning seed at 1337. The supplementary table also retains the single-seed token-matched controls and repeats finetuning with three seeds on the seed-1337 matched-time checkpoints. SST-2 and BoolQ use batch size 16, dynamic padding, and length caps of 128 and 512 tokens, respectively. The learning rate is 2×10^{-5} and AdamW weight decay is 0.01. Within each pretraining group and task, the ordinary-attention model selects an epoch from one to five on a stratified 10% internal development split; all alternatives use that epoch count and the same training examples in the same order. The official validation set is evaluated after selection. Downstream updates are matched within each seed and task, not combined pretraining and finetuning time.

The pretraining-seed and finetuning-seed comparisons share seed 1337 and are not pooled as independent repetitions.

C.3.6 Associative recall: adaptation and immediate replacement

Multi-query associative recall requires predicting the value associated with a key that appeared earlier in the sequence (Arora et al., 2024). We use the unmodified Zoology generator at revision `1ad20d1`, with symbolic vocabulary size 8,192, random non-query tokens, and `power_a`=0.01 for the distance distribution. The four matched-time models from Table 16 retain their fitted patterns or pruned layouts throughout adaptation; no heads are selected again. Each model receives 1,500 updates on 512-token sequences with eight key-value pairs, using batch size 16 and AdamW with learning rate 10^{-4} and weight decay 0.01. Loss and accuracy are measured only at supervised query positions, with all 50,304 model output classes available.

We vary three pretraining seeds with task seed 1337 fixed, then vary three task-training seeds with pretraining seed 1337 fixed. Both groups share the same seed-1337 experiment and are not pooled as six independent repetitions. The training stream is shared across methods within each task seed, with development and test data fixed across both seed axes. The ordinary-attention model must reach 80% accuracy on 256 development examples before comparing alternatives; all runs pass this predeclared check. Evaluation uses the final update rather than a test-selected checkpoint, with 512 examples per condition. The three-pretraining-seed comparison tests 8, 16, 24, 32, 48, and 64 pairs at 512 tokens, without further training, fitting, or head selection (Figure 5). All methods and seeds share

Table 17: Supplementary finetuning comparisons. Each group specifies which seed varies; errors are standard errors.

		Accuracy (%) ↑	
Operation + selection	Rate	SST-2	BoolQ
Midpoint, matched time: three pretraining seeds, one finetuning seed			
Ordinary attention	0%	86.24 ± 0.93	71.86 ± 0.40
Mean + variance	25%	86.05 ± 0.54	70.52 ± 0.16
Mean + variance	50%	86.12 ± 0.54	70.66 ± 0.26
Pruning + variance	25%	86.85 ± 0.56	70.97 ± 0.66
Pruning + variance	50%	85.93 ± 0.56	69.92 ± 0.26
Later, matched tokens: one pretraining and one finetuning seed			
Ordinary attention	0%	88.88	71.74
Mean + variance	25%	87.39	72.11
Pruning + variance	25%	87.73	71.99
Pruning + gate-Taylor	25%	87.61	71.87
Pruning from initialisation	25%	89.11	71.25
Later, matched time: one pretraining seed, three finetuning seeds			
Ordinary attention	0%	87.81 ± 0.28	71.54 ± 0.23
Mean + variance	25%	87.69 ± 0.44	71.78 ± 0.48
Pruning + variance	25%	87.31 ± 0.37	71.49 ± 0.31
Pruning + gate-Taylor	25%	87.65 ± 0.23	71.20 ± 0.03

evaluation examples, generated with seed $800001 + T + k$, where T is sequence length and k is pair count; evaluation uses micro-batch size four. Increasing the pair count also changes query placement and density under the unchanged generator. Table 18 combines this curve with the length-only tests, task-seed replication, and immediate-replacement study below.

The accuracy gap widens as more associations must be retrieved: the mean advantage over variance pruning grows from 5.65 points at 16 pairs to 25.39 points at 64 pairs. SAF exceeds both pruning variants in every pretraining seed at 24–64 pairs; at 16 pairs, seed 1338 does not share the mean advantage. The separate task-seed experiment confirms the ordering at 32 pairs, but does not evaluate the additional pair counts. The length-only tests instead favour pruning, distinguishing generalisation to more associations from generalisation to longer inputs.

To distinguish adaptation from immediate retention, we also apply all interventions to one ordinary-attention model already trained on MQAR, using the seed-1337 task checkpoint above. Calibration uses 32 held-out task sequences at the native 4K context: variance selection with the 0.2-nat compact-fitting threshold selects 36 heads, and variance pruning removes exactly those heads. Gate-Taylor selection uses the query-only loss in Equation 19. Every arm, including ordinary attention, starts fresh AdamW state because the task checkpoint contains no optimiser state, then receives the same new 512-token, eight-pair training stream at the adaptation learning rate and batch size. Table 18 includes all four test conditions immediately after intervention and after 50, 200, and 500 recovery updates. Immediate replacement does not reproduce the consistent 32-pair advantage of the fully adapted SAF models; the subsequent ordering also changes during recovery. The two studies differ in both intervention stage and calibration domain, so they distinguish the measured behaviours without isolating a single cause.

C.3.7 Training and prefill costs

Pruning removes more computation than fixed-pattern replacement because the selected heads no longer compute values or their output projections. Table 19 measures this difference at 4K, 8K, and 16K using the largest tested batch accepted by all compared methods under a 90% peak-allocation limit. Each update accumulates six micro-batches, totalling 491,520 tokens, and includes the forward pass, full-vocabulary loss, backward pass, clipping, and AdamW. Data loading, fitting, compilation, and evaluation are excluded. Pruned models are constructed from the ordinary-attention native-context checkpoints using the replaced models' head sets, solely for these systems measurements. Both methods reduce update time at each reported context, with pruning giving larger savings. The 8K and 16K pruned models are used only for timing; this comparison does not include their quality after adaptation.

Finally, Table 20 measures full-model causal prefill on the additional-training checkpoints before task finetuning. The benchmark includes the final-token vocabulary projection and uses FP32 stored

Table 18: MQAR query accuracy (%). Adaptation varies one seed axis at a time (mean ± SE); immediate replacement and recovery use one task-trained ordinary-attention parent.

				Pruning 25%	
Updates	Tokens / pairs	Ordinary attn.	SAF 25%	Variance	Gate-Taylor
Adaptation: three pretraining seeds, task seed 1337					
1,500	512 / 8	99.58 ± 0.05	99.73 ± 0.04	99.72 ± 0.03	99.62 ± 0.06
	512 / 16	81.32 ± 3.21	91.08 ± 1.58	85.43 ± 3.07	80.79 ± 4.50
	512 / 24	60.56 ± 4.22	79.76 ± 2.72	66.25 ± 3.22	60.59 ± 6.42
	512 / 32	47.54 ± 3.87	70.83 ± 4.86	52.32 ± 2.41	47.95 ± 6.06
	512 / 48	34.01 ± 3.38	61.14 ± 7.37	37.70 ± 1.54	34.49 ± 5.42
	512 / 64	26.19 ± 2.75	54.42 ± 8.70	29.03 ± 1.42	26.59 ± 4.41
	2,048 / 8	98.27 ± 0.30	97.96 ± 0.55	98.66 ± 0.11	98.65 ± 0.26
	4,096 / 8	90.35 ± 0.58	89.67 ± 1.09	90.74 ± 0.36	91.43 ± 0.60
Adaptation: pretraining seed 1337, three task-training seeds					
1,500	512 / 8	99.71 ± 0.09	99.76 ± 0.05	99.75 ± 0.05	99.72 ± 0.07
	2,048 / 8	98.01 ± 0.28	97.58 ± 0.01	98.87 ± 0.21	98.78 ± 0.19
	4,096 / 8	89.67 ± 0.32	88.21 ± 0.28	91.06 ± 0.46	90.76 ± 0.27
	512 / 32	44.55 ± 1.73	73.20 ± 4.26	52.13 ± 5.56	53.20 ± 5.25
Immediate replacement (0) and subsequent recovery updates					
0	512 / 8	99.54	99.68	99.68	99.66
	2,048 / 8	97.75	97.00	97.71	97.19
	4,096 / 8	89.21	88.75	89.48	88.79
	512 / 32	47.05	48.75	60.83	52.58
50	512 / 8	99.61	99.66	99.78	99.37
	2,048 / 8	96.88	98.27	97.61	95.61
	4,096 / 8	87.65	90.28	89.38	86.57
	512 / 32	51.93	43.45	51.54	53.65
200	512 / 8	99.68	99.46	99.83	99.17
	2,048 / 8	97.46	96.14	98.61	96.61
	4,096 / 8	88.79	87.11	90.48	87.96
	512 / 32	61.99	52.70	56.05	46.50
500	512 / 8	99.73	99.73	99.85	99.76
	2,048 / 8	97.73	97.51	97.90	97.09
	4,096 / 8	89.16	88.28	89.45	88.33
	512 / 32	44.74	58.39	56.79	48.14

Table 19: Native-context training-update costs for matched head sets. Memory changes are relative to ordinary attention at the same context and batch size.

			Update time (ms) ↓			Peak change (%)	
Context	B	Rate	Ordinary attn.	Mean	Pruning	Mean	Pruning
4K	20	25%	2172.80	2057.35	1923.26	−2.15	−2.19
4K	20	50%	2173.12	1929.10	1678.24	−3.34	−4.22
8K	10	25%	2651.80	2573.68	2307.18	−2.24	−2.88
8K	10	50%	2652.83	2408.64	1933.85	−3.27	−4.95
16K	5	25%	3714.63	3621.11	3083.03	−1.76	−5.08
16K	5	50%	3716.74	3419.69	2472.20	−2.82	−7.64

weights and prior buffers with BF16 autocast. Each alternative has four paired ordinary-attention measurements with alternating execution order, fresh model processes, ten warm-up calls, and forty timed calls. Latencies are mean synchronised wall times; speedups are mean paired ratios with SE over timing repeats. These prefill measurements use different workloads and precision settings from Table 19; their memory changes should not be interchanged. At $B = 1$, mean replacement is slower, with speedups of $0.933\times$ and $0.877\times$ for the token- and time-matched groups, respectively. All measurements concern causal prefill without token-by-token KV-cache decoding.

Table 20: Causal prefill at $B = 32$, $T = 4096$ for the single-seed 25% controls. Speedup SE is over four timing pairs.

Operation + selection	Ordinary attn. (ms)	Method (ms)	Speedup (×)	Peak change (%)
Matched-token pretraining				
Mean + variance	154.59	142.52	1.0847 ± 0.0024	$+3.75$
Pruning + variance	155.26	133.43	1.1636 ± 0.0012	-4.75
Pruning + gate-Taylor	154.22	132.63	1.1628 ± 0.0009	-5.66
Pruning from initialisation	154.27	132.64	1.1631 ± 0.0005	-4.75
Matched-time pretraining				
Mean + variance	153.29	141.34	1.0846 ± 0.0013	$+3.75$
Pruning + variance	153.55	131.91	1.1640 ± 0.0007	-4.75
Pruning + gate-Taylor	153.54	131.72	1.1657 ± 0.0011	-5.66

C.4 Finetuning the 124M checkpoints

For the 124M models, accuracy values are reported as mean ± standard error over three finetuning seeds from one pretrained checkpoint. Timing and memory measurements use three batch-order seeds at the task-specific micro-batch sizes shown below; QuALITY uses four accumulation steps for an effective batch of 16. Each timed update includes all accumulation steps, the task loss, backward pass, gradient clipping, AdamW update, and gradient reset. For paired ordinary-attention and replaced-model update times $t_{\mathrm{ord},r}$ and $t_{\mathrm{rep},r}$, finetuning speedup is $\bar{s} = \frac{1}{3}\sum_{r=1}^{3} t_{\mathrm{ord},r}/t_{\mathrm{rep},r}$; the displayed times are averaged separately. For example, the SST-2 25% speedups are 0.9633, 0.6653, and 1.0732, yielding 0.901 ± 0.122, whereas the ratio of mean times is 0.861. The corresponding replaced update times span 46.23–77.15 ms across the three timing seeds; the table retains this variation through the reported standard errors. Table 21 adds uncertainty estimates to the summary in Table 2; Figure 4b shows the paired finetuning speedups.

Table 21: Finetuning quality, update time, and peak-memory change for inherited fixed-attention checkpoints.

Task / checkpoint	B	Accuracy (%)	Δ accuracy (pp)	Time/update	Speedup	Peak memory change
SST-2, ordinary attn.	256	86.39 ± 0.27	—	50.11 ± 0.61 ms	$1.000\times$	0.00%
SST-2, fixed 25%	256	86.54 ± 0.36	$+0.15 \pm 0.53$	58.22 ± 9.58 ms	$0.901 \pm 0.122\times$	-3.65%
SST-2, fixed 50%	256	86.77 ± 0.74	$+0.38 \pm 1.00$	56.70 ± 9.43 ms	$0.925 \pm 0.123\times$	-5.95%
BoolQ, ordinary attn.	64	71.40 ± 0.22	—	67.36 ± 0.51 ms	$1.000\times$	0.00%
BoolQ, fixed 25%	64	70.62 ± 0.55	-0.77 ± 0.56	70.91 ± 7.25 ms	$0.967 \pm 0.083\times$	-4.63%
BoolQ, fixed 50%	64	70.84 ± 0.49	-0.56 ± 0.70	69.35 ± 7.61 ms	$0.992 \pm 0.093\times$	-7.19%
QuALITY 16K, ordinary attn.	4	26.59 ± 0.44	—	578.72 ± 2.16 ms	$1.000\times$	0.00%
QuALITY 16K, fixed 25%	4	26.09 ± 0.40	-0.50 ± 0.63	552.90 ± 0.90 ms	$1.047 \pm 0.005\times$	-2.12%
QuALITY 16K, fixed 50%	4	26.94 ± 0.62	$+0.35 \pm 0.18$	517.88 ± 3.71 ms	$1.118 \pm 0.010\times$	-3.92%

Table 22 reports results when replacement is applied during finetuning rather than inherited from pretraining. One-time replacement incurs a modest one-off cost but does not improve accuracy or update time. The train-loss-triggered schedule repeatedly measures attention and is especially costly on BoolQ. Post-training replacement avoids training-time kernel overhead but provides no adaptation after replacement.

Zero-shot evaluations on HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2019), ARC-Easy (Clark et al., 2018), SST-2 (Socher et al., 2013), and BoolQ (Clark et al., 2019a) find no Holm-corrected significant change after 25% replacement; this does not establish performance equivalence.

Table 22: Replacing 25% of heads under three RoPE finetuning schedules.

Task	Schedule	Δ accuracy (pp)	Time/update	Controller cost
SST-2	One-time	-0.19 ± 0.40	33.27 ms	0.70 s
	Train-loss	-0.65 ± 0.77	36.15 ms	45.12 s
	Post-training	-1.03 ± 0.24	30.75 ms	0.71 s
BoolQ	One-time	-1.27 ± 1.29	40.85 ms	6.18 s
	Train-loss	-0.38 ± 0.15	66.34 ms	70.15 s
	Post-training	-1.03 ± 0.33	31.44 ms	6.17 s

C.5 1B TRAINING AND EVALUATION

Pretraining and perplexity. The 32-layer decoder has 16 attention heads per layer, hidden dimension 1,536, an MLP expansion factor of four, tied token embeddings, and no biases or dropout. It contains 983,336,448 trainable parameters; an unused frozen absolute-position table retained for checkpoint compatibility brings the stored parameter count to 995,919,360. RoPE is used throughout, with base 10,000 and context length 8,192. Four GH200 GPUs each use micro-batch size two and eight accumulation steps, giving 524,288 target tokens per optimiser update without activation checkpointing. Training performs 37,512 updates, processing 19,667,091,456 target tokens from a shuffled non-overlapping FineWeb-Edu corpus without repeating target positions. AdamW uses a peak learning rate of 3×10^{-4}, 375 linear warm-up updates, cosine decay to 3×10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.1, and gradient clipping at norm 1.

All four configurations share pretraining seed 1337 and the update-18,756 midpoint checkpoint, including optimiser and sampler state. Replacement selects 128 or 256 of the 512 heads by attention variance and fits their post-softmax means with the absolute-plus-relative representation. Calibration samples 16 sequences of length 8,192 from the training corpus and measures the current checkpoint on rank zero without updating its weights or advancing the training sampler; fitting uses 400 steps and the same 0.2-nat mean row-wise KL acceptance threshold as the 124M experiments. Table 13 reports calibration, fitting, and total intervention time. The pruning control applies the gate-Taylor criterion in Equation 19 to the same 16 calibration sequences and removes the 128 lowest-scoring heads. It physically removes the corresponding projection rows and columns, retaining the remaining AdamW state, learning-rate schedule, and data order through update 37,512. Final perplexities in Table 3 use 512 sequential 8K windows, or 4,194,304 target tokens, from a reporting interval disjoint from calibration. Pretraining results use one seed; the downstream evaluations use three finetuning seeds for each resulting checkpoint.

Post-replacement update measurements. Table 23 distinguishes four-GPU update time, including gradient communication, from single-GPU time and memory. The distributed benchmark uses the training configuration: four GH200 GPUs, $B = 2$ per rank, eight accumulation steps, and 524,288 tokens per update. Four pairs per replacement rate alternate execution order; each block reloads the source weights, starts fresh benchmark AdamW state, and performs ten warm-up and five measured updates. Only one model is resident per block, and all models use identical preloaded native-length batches. Update time is the slowest rank's wall time, including forward and backward computation, clipping, AdamW, and gradient all-reduce, but excluding loading, calibration, warm-up, and checkpoint I/O. Reported times are means across blocks; speedups are means of within-pair ratios with SE across the four pairs, not pretraining seeds.

The single-GPU benchmark uses four fresh-process pairs, two in each execution order, with three warm-up and ten timed optimiser updates per model in each pair. At $B = 1$ and $B = 2$, accumulation counts of 16 and eight keep the workload at 131,072 tokens per update, matching the per-rank pretraining workload. Ordinary-attention heads use forced FlashAttention, and fused dispatch is verified for replaced heads. Reported times are medians across pairs; speedups are medians of within-pair ratios, and peak-memory changes compare the paired configurations at the same batch size. Separate ordinary-attention measurements accompany the two replacement rates. These measurements include gradient accumulation and the optimiser update but exclude data loading, distributed communication, calibration, and pattern fitting.

Table 23: 1B post-replacement update cost at 8K. Four-GPU speedup errors are SE over four timing pairs; single-GPU results use medians.

GPUs / B	Rate	Ordinary attn. (ms)	SAF (ms)	Speedup (×)	Peak change (%)
Four GPUs, including communication: 524,288 tokens per update					
4 / 2	25%	4258.67	3986.67	1.0682 ± 0.0005	—
4 / 2	50%	4263.01	3653.78	1.1667 ± 0.0003	—
Single GPU, no communication: 131,072 tokens per update					
1 / 1	25%	4224.85	3958.26	1.067	−0.57
1 / 1	50%	4226.87	3632.25	1.163	−2.31
1 / 2	25%	4081.61	3828.21	1.066	−1.83
1 / 2	50%	4083.50	3505.94	1.165	−3.99

Downstream finetuning. Table 3 summarises quality for the four pretrained checkpoints; Table 24 gives individual finetuning seeds, paired changes, and observed update times. Training uses causal-LM answer-token supervision with learning rate 2×10^{-5}, AdamW weight decay 0.01, gradient clipping at norm 1, and bfloat16. For each task and seed, the ordinary-attention model selects an epoch from one through five on a stratified 10% internal development split; all alternatives train for exactly that many epochs on the same examples in the same order before evaluation on the official validation set. Selected epochs are 5/1/1 for SST-2, 4/2/4 for BoolQ, and 5/5/5 for QuALITY. The pretrained head masks, patterns, and pruned layouts are inherited unchanged, with no additional selection or fitting. SST-2, BoolQ, and QuALITY use token caps of 128, 512, and 8,192, and micro-batches of 128, 32, and four, respectively. Only QuALITY accumulates gradients over four micro-batches; mean training-example lengths are 25.4, 142.8, and 6,207.5 tokens.

The paired accuracy differences between SAF 25% and gate-Taylor pruning are -0.08 ± 0.04, -0.19 ± 2.07, and -0.06 ± 0.75 percentage points on SST-2, BoolQ, and QuALITY, respectively. These three finetuning repetitions measure task-training variation, not equivalence between methods or variation across pretraining seeds.

We measure 1B finetuning time within the training loop up to the selected epoch, including batching and optimisation and excluding development evaluation. Each seed contributes its mean time per optimiser update; the table reports the mean and standard error across seeds. Runs use different GPU allocations: all pruning runs and most 25% replacement runs reuse previously completed ordinary-attention references. Pruning, 25% replacement, and two QuALITY seed groups use `expandable_segments:True`; earlier runs use the default allocator. Successful replacements for two out-of-memory QuALITY attempts retain the same context, batch size, and task protocol. The table therefore reports observed training times rather than controlled paired speedups; Table 23 uses controlled pairs for pretraining-update costs.

Table 24: 1B finetuning by seed: accuracy (%), paired change from ordinary attention (pp), and observed mean update time. Errors are SE across finetuning seeds.

		Accuracy by seed ↑				
Task	Model	1337	1338	1339	Change (pp)	Update (ms)
SST-2	Ordinary attention	91.63	90.94	91.74	—	173.58 ± 0.74
	SAF 25%	90.94	91.86	92.09	$+0.19 \pm 0.47$	166.78 ± 2.75
	Pruning 25%	91.06	91.86	92.20	$+0.27 \pm 0.44$	156.60 ± 0.97
	SAF 50%	91.28	91.51	90.14	-0.46 ± 0.63	158.36 ± 3.67
BoolQ	Ordinary attention	75.99	76.39	75.57	—	222.70 ± 0.32
	SAF 25%	72.39	77.03	76.39	-0.71 ± 1.45	215.11 ± 0.39
	Pruning 25%	75.69	73.30	77.40	-0.52 ± 1.43	200.03 ± 0.74
	SAF 50%	75.17	76.24	75.63	-0.31 ± 0.27	196.13 ± 0.59
QuALITY	Ordinary attention	24.98	28.33	29.00	—	3504.01 ± 5.63
	SAF 25%	27.71	27.90	27.18	$+0.16 \pm 1.35$	3267.23 ± 10.13
	Pruning 25%	26.27	28.62	28.09	$+0.22 \pm 0.64$	2972.08 ± 1.10
	SAF 50%	26.32	23.73	25.31	-2.32 ± 1.85	2989.63 ± 5.40

C.6 REPLACEMENT IN PRETRAINED MODELS

Qwen3-4B has 36 layers, 32 query heads, eight key/value heads, and head dimension 128.

C.6.1 QWEN3-4B ZERO-SHOT TASK ACCURACY

We evaluate the five zero-shot tasks using loglikelihood over the full validation split of each task, comparing candidate continuations without generation and grouping requests by exact token length so that no padding enters the fused kernel. Both selectors use the same fitting protocol and 512 FineWeb-Edu calibration sequences, with compact patterns fitted separately for their selected head sets. The median row-wise fit KL values are similar, at 0.027 nats for variance-selected heads and 0.029 nats for forward-KL-selected heads.

Table 25: Qwen3-4B zero-shot accuracy (%) with compact FineWeb-Edu mean patterns. The last row gives the mean change from the ordinary-attention model in percentage points.

		Attention variance			Forward KL		
Task	Ordinary attn.	10%	20%	30%	10%	20%	30%
HellaSwag	52.28	52.02	51.21	48.92	49.10	45.05	43.81
PIQA	74.86	75.19	74.48	73.56	74.70	72.58	70.40
ARC-Easy	80.43	79.63	77.61	75.72	72.90	68.06	64.10
SST-2	88.65	88.65	87.96	85.44	87.61	63.07	68.81
BoolQ	84.46	83.33	80.49	73.12	80.03	71.38	71.38
Mean change	—	−0.37	−1.79	−4.79	−3.27	−12.11	−12.44

Variance selection gives higher accuracy in all fifteen task–rate comparisons in Table 25. The selected sets overlap in 31% of heads at 10% replacement and 43% at 30%. At 10%, the counts over layers 1–12, 13–24, and 25–36 are 50/18/47 for forward KL and 37/31/47 for variance. The variance-selected heads also have a higher median per-input forward-KL score, 0.578 versus 0.296 nats at 10% replacement, despite giving higher task accuracy when replaced together. These observations show that the scores select different head sets; neither the layer counts nor the pattern distances establish which head functions cause the accuracy differences.

SST-2 accuracy under forward-KL selection falls by approximately 25.6 percentage points at 20% replacement, then recovers partially at 30%. The prediction counts for its two labels change from 479/393 in the ordinary-attention model to 721/151 at 20% and 580/292 at 30%, compared with gold counts of 428/444. Thus, the largest accuracy loss coincides with the strongest imbalance towards one predicted label, but these aggregate counts do not establish the mechanism behind the non-monotone curve. Under variance selection, the SST-2 accuracy loss is at most 3.21 points across the tested rates.

C.6.2 GQA PREFILL MEASUREMENTS

The prefill benchmark compares the fused GQA path with Flash scaled dot-product attention at $B \in \{4, 8\}$ and $T \in \{2048, 4096, 8192\}$. It uses forward-KL-selected heads and patterns fitted on 128 FineWeb-Edu sequences of 2,048 tokens. Table 26 reports end-to-end and attention-operation speedups for these systems configurations, separately from the 512-sequence task evaluation above. Measurements beyond the calibration length assess execution speed.

Table 26: Qwen3-4B prefill speedup (×): end-to-end, with attention-operation speedup in parentheses.

B	T	10%	20%	30%	50%	75%
4	2048	1.011 (1.16)	1.028 (1.18)	1.068 (1.21)	1.135 (1.29)	1.225 (1.38)
4	4096	1.027 (1.16)	1.057 (1.19)	1.097 (1.23)	1.156 (1.28)	1.246 (1.38)
4	8192	1.042 (1.15)	1.072 (1.19)	1.115 (1.23)	1.184 (1.30)	1.250 (1.37)
8	2048	1.015 (1.15)	1.042 (1.17)	1.079 (1.21)	1.148 (1.28)	1.231 (1.37)
8	4096	1.029 (1.15)	1.061 (1.19)	1.080 (1.18)	1.156 (1.27)	1.252 (1.37)
8	8192	1.040 (1.15)	1.071 (1.18)	1.110 (1.22)	1.178 (1.30)	1.272 (1.40)

D KERNEL IMPLEMENTATION AND VALIDATION

D.1 COMPACT FITTING AND STORAGE

The need for compact storage is particularly clear at long context lengths: at $T = 10^6$, one causal fp32 matrix occupies approximately 2 TB even if only its lower triangle is stored. Equation 4 is the maximum-likelihood objective of a log-linear model and is convex in (α, ρ) up to additive shifts absorbed by row normalisation. At an optimum, the fitted prior reproduces both target marginals because

$$\frac{\partial \mathcal{L}_{\text{fit}}}{\partial \alpha(j)} = \frac{1}{T}\left(\sum_i \widehat{\mathbf{P}}(i,j) - \eta_{\text{abs}}(j)\right), \tag{20}$$

with the analogous condition for ρ. The fitting settings and acceptance threshold are given in Appendix C.1. The dense implementation explicitly forms causal rows; the FFT implementation evaluates the convolutional row normalisers from the sufficient statistics. These fitting paths are distinct from reconstruction inside the fused execution kernel. For the 4K RoPE head set in Table 27, the dense implementation allocates an fp32 $H \times T \times T$ buffer in each of the ten affected layers, including unused head slots, for 8.060 GB in total. The compact buffers occupy 0.133 GB; an ideally packed dense representation containing only the 36 selected matrices would occupy 2.416 GB.

D.2 MIXED-HEAD EXECUTION AND NUMERICAL CHECKS

The fused kernel executes ordinary-attention and replaced heads in a single forward launch per layer, writing outputs in their original head order. Each kernel program handles one batch element, head, and block of query positions; a head-state flag selects the computation for that program. Ordinary-attention heads use the online-softmax tiling of FlashAttention-2 (Dao et al., 2022; Dao, 2024). For a replaced head, the program reconstructs each causal tile of $\widehat{\mathbf{P}}$ in registers from α, ρ, and precomputed row normalisers, multiplies it with the value tile, and accumulates the output. The fixed path neither computes query-key scores nor reads a dense $T \times T$ pattern, and preserving head order avoids concatenating separate ordinary-attention and replaced-head outputs.

For fixed attention weights, the attention operation's backward pass is $\mathrm{d}\mathbf{V} = \widehat{\mathbf{P}}^\top \mathrm{d}\mathbf{Z}$. Key-block programs accumulate value gradients for every head and key gradients for ordinary-attention heads; a separate query-block pass computes query gradients only for ordinary-attention heads. Our multi-head pretraining implementation computes query and key projections only for ordinary-attention heads. The Qwen grouped-query implementation omits replaced query projections but retains the full shared key and value projections.

Persistent state per replaced head consists of α, ρ, the normalisers, and a small aligned relative-distance band for coalesced tile loads, using $O(T)$ storage. The multiplication $\widehat{\mathbf{P}}\mathbf{V}$ still costs $O(T^2 d_h)$: savings come from removing score and softmax computation and reducing pattern bandwidth and activation storage.

Table 27 combines pattern fidelity, storage, and forward checks; Table 28 reports gradient errors against the eager reference.

Table 27: Representation quality at 25% replacement (one seed), prior storage, and kernel checks.

Position	Ordinary attn. PPL	Dense PPL	Abs.+rel. PPL	Storage reduction
Absolute	26.0252	26.1184	26.1150	59.9×
RoPE	23.9571	24.1180	24.1165	60.6×

Component	Configuration	Metric	Result
Prior buffers	RoPE, 36 heads in 10 layers	Dense impl. / abs.+rel.	8.060 / 0.133 GB
Mixed-head forward	nanoGPT, FP32	Maximum absolute error	1.07×10^{-6}
State transitions	50-operation fuzz test	Worst oracle error	9.54×10^{-7}
Mixed-head backward	Replaced / ordinary-attention Q,K rows	Gradient status	zero / nonzero
Fused dispatch	nanoGPT, BF16	Triton calls	2 / 2

We check the fused implementation against an eager reference in fp32 and bfloat16. The suite covers empty, partially replaced, and fully replaced head sets; unaligned sequence lengths; non-contiguous query/key views; head dimensions 64, 96, and 128; and mixtures of dense and absolute-plus-relative patterns. Replaced heads have no query/key slot on the multi-head long-context path and therefore receive zero query/key gradients by construction. The grouped-query path maps each ordinary-attention query head to its retained shared key/value group. Table 28 reports numerical errors for a mixed-head RoPE layer on GH200, comparing fused execution with an eager dense reference that reconstructs the same stored compact patterns. The test uses batch size two, sequence length 129, four heads of dimension 64, and a stored prior length of 192; heads 1 and 3 are fixed, using zero-based indices. Both paths receive the same output gradient, and fused dispatch is checked. Parameters remain FP32; the BF16 test uses autocast, as in training. For reference tensor u and fused result $\widetilde{u}$, we report maximum absolute error and relative L_2 error, $\|\widetilde{u} - u\|_2 / \max(\|u\|_2, 10^{-8})$. The query/key weight comparisons include the ordinary-attention heads' gradients and the zero rows of replaced heads; value gradients include both head types. The same seven-component test for pruning gives maximum relative L_2 errors of 7.48×10^{-7} in FP32 and 4.45×10^{-3} under BF16 autocast.

Table 28: Numerical agreement with the eager reference for a layer with ordinary-attention and fixed-mean heads. Projection rows report weight gradients.

	FP32		BF16 autocast	
Tensor	Max. absolute	Relative L_2	Max. absolute	Relative L_2
Output	1.19×10^{-7}	1.86×10^{-7}	3.91×10^{-3}	2.80×10^{-3}
Input gradient dX	3.58×10^{-7}	2.51×10^{-7}	7.81×10^{-3}	3.68×10^{-3}
Value gradient dV	3.58×10^{-7}	1.78×10^{-7}	7.81×10^{-3}	1.46×10^{-3}
Query projection	8.34×10^{-7}	2.82×10^{-7}	1.56×10^{-2}	4.53×10^{-3}
Key projection	1.19×10^{-6}	3.01×10^{-7}	1.56×10^{-2}	4.43×10^{-3}
Value projection	3.81×10^{-6}	2.30×10^{-7}	6.25×10^{-2}	2.05×10^{-3}
Output projection	3.81×10^{-6}	1.97×10^{-7}	6.25×10^{-2}	2.68×10^{-3}

D.3 CAUSAL-PREFILL BENCHMARKS

The checkpoint-backed prefill benchmark compares the seed-1337 ordinary-attention 16K RoPE checkpoint under forced FlashAttention with its 25%- and 50%-replaced counterparts. Inputs are seeded token tensors, and each timing point is the median of four or six fresh-process ABBA pairs without CUDA-Graph replay. The input-length sweep fixes $B = 64$ over $T \in \{512, 1024, 2048, 4096, 8192, 16384\}$, while the batch-size sweep fixes $T = 4096$ over $B \in \{2, 4, 8, 16, 32, 64\}$. A separately collected launch-bound $B = 1$ measurement is retained only as a diagnostic.

Table 29: Selected causal-prefill latencies for the ordinary-attention and replaced 16K RoPE checkpoints at $B = 64$, without CUDA Graph replay. Input length varies while the checkpoints remain fixed.

Input length	Batch size B	Replaced heads	Ordinary attn. (ms)	Replaced (ms)	Speedup
4K	64	25%	306.70	278.33	1.100×
4K	64	50%	303.94	247.47	1.229×
8K	64	25%	776.17	705.03	1.100×
8K	64	50%	760.20	622.94	1.221×
16K	64	25%	2075.04	1900.13	1.092×
16K	64	50%	2045.38	1703.87	1.200×

Figure 4c,d shows the checkpoint-backed causal-prefill speedups across input lengths and batch sizes; Table 29 gives selected absolute latencies. Latency columns are medians across pairs, whereas speedups are medians of within-pair ratios and need not equal the ratios of the displayed latency medians. The figure's whiskers span the observed pairs. At fixed batch size 64, the checkpoint-backed benchmark remains above break-even over the complete 512–16K input-length range. The measured ranges are 1.092–1.116× at 25% replacement and 1.200–1.244× at 50%. At fixed 4K length, the primary batch sweep uses sizes two through 64 and is similarly stable.

The separately retained batch-one point is dominated by launch overhead and exhibits large process-to-process variation. At $T = 4096$ and $B = 1$, the median paired speedups are 0.969× and 0.972× at 25% and 50% replacement, respectively; their observed ranges are 0.878–1.067× and 0.599–1.498× across six pairs.

We retain measurements only when the zero-replacement control remains within the contention tolerance of 1.00×.

In batch-one shape benchmarks, the 124M shape at 16K reaches 1.08× causal-prefill speedup with 25% replacement and 1.33× with full replacement. For a 774M shape, the 25% speedup is 1.07× at 8K and 1.08× at 16K, while full replacement reaches 1.31× and 1.36×. CUDA-Graph replay isolates launch overhead for the checkpoint-backed 4K model: at batch size one, 25%, 50%, and 75% replacement reach 1.113×, 1.220×, and 1.377× speedup, respectively. These are forward-only kernel measurements. They do not measure model perplexity, token-by-token decoding, K-cache removal, or static query/key parameter repacking.

E Data, Models, and Licences

Table 30 lists the external datasets and pretrained weights used in the reported experiments, including the auxiliary zero-shot evaluations in Appendix C.4. Asset names link to the repositories from which they can be accessed; licence entries link to the corresponding upstream statements. We train the 124M models in the controlled pretraining study from scratch.

Table 30: External datasets and pretrained models.

Asset	Use	Upstream licence or terms
Datasets		
FineWeb-Edu	Pretraining; calibration and evaluation	ODC-By 1.0; Common Crawl terms
SST-2 (GLUE)	Finetuning and zero-shot evaluation	Not specified in the original dataset card
BoolQ (SuperGLUE)	Finetuning and zero-shot evaluation	CC BY-SA 3.0
QuALITY	Long-input finetuning	CC BY 4.0; article-level licences
HellaSwag	Zero-shot evaluation	MIT
PIQA	Zero-shot evaluation	Academic Free License 3.0
ARC-Easy	Zero-shot evaluation	CC BY-SA 4.0
Pretrained weights		
Qwen3-4B	Static replacement and prefill	Apache 2.0

The FineWeb-Edu database licence does not replace the rights in the underlying web content; the release also refers users to Common Crawl's terms of use. QuALITY records a separate licence for each source article in its upstream data. GLUE directs users to the original dataset licences; its loading code does not establish a licence for SST-2's review text.

The implementation builds on nanoGPT and Triton, both under MIT licences. The 124M experiments use the GPT-2 tokeniser through tiktoken (MIT); the Qwen experiments use the tokeniser distributed with Qwen3-4B. The synthetic MQAR data use the Zoology generator, distributed under Apache 2.0.