FREQUENCY-AWARE ATTENTION: TOKENIZER-INDUCED SELF-INFORMATION AS AN ARCHITECTURAL INDUCTIVE BIAS

Anonymous authors
Paper under double-blind review

ABSTRACT

The performance of Transformer-based large language models (LLMs) is heavily influenced by the power-law token-frequency distribution induced by the tokenizer and corpus, leading to disparate learning dynamics in which LLMs struggle with long-tail tokens. Prior approaches inject token-frequency statistics through data, loss, output-logit, tokenizer, or embedding interventions; we instead make these statistics explicit as an architectural prior within Transformer attention. We introduce **Frequency-Aware Attention (FAA)**, which injects corpus-derived self-information—defined as the negative log-probability of tokens—as an explicit prior into the attention mechanism in the form of attention-logit bias, value scaling, and gating. At the 150M scale, gate-free FAA—using only attention-logit bias and value scaling—reduces perplexity (PPL) by 3.1% and rarest-decile perplexity (D0 PPL) by 7.2% in a representative run, with only 0.043% additional parameters. Full FAA reduces PPL by 4.9% and D0 PPL by 8.9% on average across three seeds, with approximately 4.9% additional parameters. It also improves ARC-E and ARC-C accuracy by 1.89 and 2.05 percentage points, respectively. Our results suggest that making token-frequency statistics architecturally explicit yields a useful inductive bias for language modeling; PPL reductions persist at 450M and 1.2B.

1 INTRODUCTION

Modern large language models (LLMs) acquire their capabilities through pre-training on massive text corpora (Brown et al., 2020; Radford et al., 2019). However, this data-driven paradigm inherits fundamental statistical properties of natural language—most notably, the imbalanced token distribution characterized by Zipf's law (Zipf, 1949). A growing body of work characterizes how this imbalance induces disparate learning dynamics across tokens: proficiency on specific words and concepts correlates strongly with their training-set frequency (Chang & Bergen, 2022; Kandpal et al., 2023; Razeghi et al., 2022; Wei et al., 2021); separately, representation degeneration yields anisotropic embedding spaces (Gao et al., 2019), while rare-token degeneration can degrade performance beyond rare tokens themselves (Yu et al., 2022).

Throughout the paper, the *corpus-derived unigram prior* denotes the tokenizer-indexed table $P(t)$ obtained by applying the final tokenizer to the corpus and counting token occurrences. We inject this prior into Transformer attention as an *extrinsic statistical signal*—supplied externally from corpus statistics rather than inferred from model parameters—aiming to improve rare-token learning without degrading high-frequency token performance. We propose **Frequency-Aware Attention (FAA)**, which injects a precomputed tokenizer-indexed unigram prior into three attention-block pathways: (1) Frequency-Aware Bias (FAB), which adds a key-side unigram prior to content-based attention logits, complementing the learned query–key interaction with an explicit frequency bias; (2) Information-Weighted Value Scaling (IWVS), which rescales value vectors by a learned function of self-information, motivated by analogy with smooth inverse frequency (SIF) sentence-embedding aggregation (Arora et al., 2017)—which downweights frequent words—and by prior analyses showing that attention outputs depend on transformed value magnitudes in addition to attention weights (Gupta & Berant, 2021; Kobayashi et al., 2020); and (3) Frequency-Aware Information Gate (FAIG), which adds a frequency-conditioned offset to the content-dependent information gate (IG) of Gated Atten-

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.

tion (Qiu et al., 2025). Together, these pathways let a single external unigram table affect routing, value magnitudes, and output gating inside each attention block.

We summarize our contributions as follows:

- We propose **FAA**, which injects a tokenizer-indexed unigram prior into Transformer attention through key-side attention-logit bias, value scaling, and output-gate modulation.
- We separate frequency-specific effects from the inherited gate using gate-free FAA (FAB+IWVS; 0.043% additional parameters), gate-matched ablations, shuffled/random priors, shared learned-token controls, and frequency-aware baselines outside attention.
- We show consistent perplexity (PPL) and rarest-decile perplexity (D0 PPL) reductions at 150M, 450M, and 1.2B scales, with cross-architecture and PubMed-domain transfer.

2 Related work

Prior interventions that address frequency or imbalance operate through data-level balancing (Zevallos et al., 2023), loss-level class-imbalance reweighting (Cui et al., 2019), output-logit adjustment (Menon et al., 2021), tokenizer changes (Lian et al., 2025), or embedding-level debiasing. At the embedding level, Frequency-Agnostic Word Representation (FRAGE) (Gong et al., 2018) uses adversarial training to reduce frequency-related separability; output-head analyses separately show that Transformer prediction heads encode word-frequency effects (Kobayashi et al., 2023). **FAA** instead keeps the tokenizer fixed and injects corpus frequency *within the attention block*, where routing weights, value magnitudes, and output gates control how contextual evidence flows through the residual stream. This design complements value-path analyses (Gupta & Berant, 2021; Kobayashi et al., 2020), the Gated Attention backbone (Qiu et al., 2025), and positional priors: RoPE (Su et al., 2024), Attention with Linear Biases (ALiBi) (Press et al., 2022), and T5-style relative position bias (Raffel et al., 2020) encode *where* tokens appear or how far apart they are, whereas FAA supplies a tokenizer-indexed lexical prior about *which* vocabulary item appears. Frequency-Aware Sparse Attention (FASA) (Wang et al., 2026) predicts salient key-value (KV) positions from RoPE frequency chunks for runtime long-context sparsification; FAA keeps dense attention and injects a static corpus-unigram prior into the attention-logit, value, and gate pathways. The unigram prior is also motivated by learning dynamics: neural language models rely strongly on unigram frequencies early in training before gradually transitioning toward richer contextual predictions (Chang & Bergen, 2022), and recent work argues that token-frequency structure itself can be useful for pre-training (Chung & Kim, 2025; He et al., 2025). We therefore treat corpus unigram probabilities as an external statistic made explicit where attention allocates and transforms information. Full discussion of adjacent sparse-attention and tokenizer-side work appears in Appendix J.

3 Frequency-aware attention

Notation and backbone. We use 0-based indexing for positions $i, j \in \{0, \dots, T-1\}$, heads $h \in \{0, \dots, H-1\}$, and layers $\ell \in \{0, \dots, L-1\}$. Standard multi-head attention (Vaswani et al., 2017) projects input $X \in \mathbb{R}^{T \times d_{\text{model}}}$ into per-head Q_h, K_h, V_h and computes

$$A_h = \text{softmax}\Big(\frac{Q_h K_h^\top}{\sqrt{d_k}}\Big), \qquad \text{head}_h = A_h V_h, \tag{1}$$

with row-wise softmax under causal masking. Gated Attention (Qiu et al., 2025) concatenates the head outputs and applies an elementwise IG before the output projection:

$$Z = \text{Concat}(\text{head}_0, \dots, \text{head}_{H-1}) \in \mathbb{R}^{T \times H d_v},$$
$$W_{\text{IG}} \in \mathbb{R}^{d_{\text{model}} \times H d_v}, \qquad G_{\text{IG}} = \sigma(X W_{\text{IG}}), \qquad O = (Z \odot G_{\text{IG}}) W^O.$$

Here, σ is the sigmoid function; $H d_v = d_{\text{model}}$, so $Z, G_{\text{IG}} \in \mathbb{R}^{T \times d_{\text{model}}}$ and $W^O \in \mathbb{R}^{H d_v \times d_{\text{model}}}$. The gate controls how much aggregated context is written to the residual stream, with one logit per head output channel. Full FAA extends IG with a frequency-conditioned offset to these content logits, yielding FAIG (Eq. (7)). A headwise variant uses $W_{\text{IG}} \in \mathbb{R}^{d_{\text{model}} \times H}$ and broadcasts across d_v (Appendix I.1). Bias terms are suppressed; fused query-key-value (QKV)–gate bias accounting is in Appendix C. We suppress the layer index ℓ on per-layer parameters (α_h, γ_h, β) except in cross-layer comparisons.

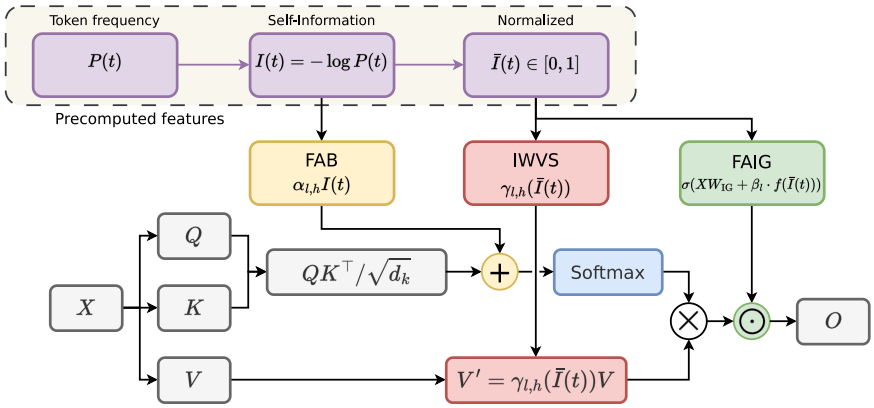


Figure 1: **Overview of FAA.** The corpus unigram prior enters attention through FAB key-side attention-logit bias, IWVS value scaling, and FAIG output-gate modulation.

3.1 The statistical prior: self-information

The fundamental signal for FAA is token self-information (Shannon, 1948):

$$I(t) = -\log P(t) \tag{2}$$

where $P(t)$ is obtained by applying the final tokenizer to the prior-estimation corpus and counting token occurrences.[1] Self-information increases as token frequency decreases, giving each vocabulary item a scalar rarity signal precomputable from unigram counts in an $O(|\mathcal{V}|)$ table. Reassigning these values by vocabulary ID or character entropy yields similar PPL but lower ARC-E/ARC-C accuracy (Appendix H.2, Table 20). Although language models ultimately estimate $P(t \mid c)$, surprisal theory treats context-conditional negative log-probability as a quantitative model of token-level processing difficulty (Hale, 2001; Levy, 2008). Our unigram quantity $I(t) = -\log P(t)$ is only a precomputed marginal proxy for token rarity, not a substitute for runtime conditional surprisal. Prior work shows that neural language models rely on unigram frequencies early in training before transitioning toward bigram and richer predictions (Chang & Bergen, 2022). This motivates supplying the marginal statistic explicitly while leaving context-dependent refinement to learned attention parameters. Complementary mechanistic analyses of decoder-only language models identify final-layer "token frequency neurons" whose effects move output distributions toward or away from empirical unigram statistics (Stolfo et al., 2024). This suggests that unigram frequency is already used implicitly inside trained language models; FAA instead supplies a corpus-derived unigram prior explicitly inside attention.

Zero-count and extreme-tail tokens are handled with quantile floor clamping, and special tokens use fixed reference statistics rather than corpus-frequency estimates. Appendix B gives the exact clamping rule, special-token overrides, and sensitivity analysis.

3.2 FAA architecture

FAA introduces three complementary components into multi-head self-attention as shown in Figure 1. The components occupy distinct axes of the attention block: FAB is key-side (which tokens are retrieved), IWVS is value-side (how strongly retrieved evidence contributes), and FAIG is query-side (how much aggregated context is written back at position i).

Frequency-aware bias (FAB). FAB adds a static key-side unigram prior with a learned per-head strength to the content-based attention logits. For each layer–head pair, a signed scalar α_h modulates the self-information of the key token:

$$s_{hij} = \frac{\mathbf{q}_{hi}^\top \mathbf{k}_{hj}}{\sqrt{d_k}} + \alpha_h \, I(t_j) \tag{3}$$

[1] We use natural logarithms.

where t_j is the key-token ID and $I(t_j)$ is broadcast across query positions. Setting $\alpha_h{=}0$ recovers Eq. (1), so the FAB bias path is initialized neutrally (Appendix D). A positive α_h favors high-self-information keys, whereas a negative α_h favors high-frequency keys. Because the added term depends only on key identity, FAB changes the routing distribution while leaving the query–key dot product unchanged; a query-side analogue $\alpha_h I(t_i)$ would add a row-wise constant and vanish under the softmax, motivating the key-side placement.

Since $I(t_j){=}-\log P(t_j)$, FAB equivalently reweights each visible key's unnormalized attention weight by a learned power of its unigram probability:

$$A_{hij} = \frac{\exp\left(\frac{\mathbf{q}_{hi}^\top \mathbf{k}_{hj}}{\sqrt{d_k}}\right) P(t_j)^{-\alpha_h}}{\sum_{m \le i} \exp\left(\frac{\mathbf{q}_{hi}^\top \mathbf{k}_{hm}}{\sqrt{d_k}}\right) P(t_m)^{-\alpha_h}}, \qquad j \le i. \tag{4}$$

This makes FAB a learned token-frequency power reweighting of key positions, distinct from output-logit adjustment (Menon et al., 2021) and ALiBi's distance-only positional bias (Press et al., 2022); Appendix G gives the extended comparison and motivation from frequency-sensitive attention heads.

Information-weighted value scaling (IWVS). While FAB reshapes routing, IWVS leaves attention weights unchanged and rescales value vectors before aggregation. Its motivation is value-path modulation: SIF sentence embeddings aggregate word vectors with frequency-conditioned weights $a/(a+p(w))$ (Arora et al., 2017). IWVS borrows this aggregation-side intuition—that per-token weighting at the aggregation step alters the resulting representation—rather than SIF's monotone downweighting rule. In causal language-model pre-training, attenuation is already supplied by the row-normalized attention distribution and the IG/FAIG output gate; IWVS therefore uses a learned amplification factor $\gamma_h(t) > 1$ for non-special tokens, and a bidirectional variant that permits downscaling does not improve PPL (Appendix H.2, Table 20). Value-path analyses support intervening on this pathway: BERT transformed-value norms correlate with token-frequency rank (Spearman $\rho{\approx}0.69$ after excluding special/punctuation tokens) (Kobayashi et al., 2020), and sparse-attention approximations that account for value vectors recover attention outputs more accurately than weight-only approximations (Gupta & Berant, 2021).

IWVS maps min-max normalized self-information $\bar{I}(t) \in [0,1]$ to one scale per head using a two-layer multilayer perceptron (MLP) in each layer; special tokens map to the neutral reference value:

$$\bar{I}(t) = \begin{cases} 0, & t \in \mathcal{S} \\ \frac{I(t)-I_{\min}}{I_{\max}-I_{\min}}, & \text{otherwise} \end{cases} \tag{5}$$

Here, $I_{\min}$ and $I_{\max}$ are computed over non-special vocabulary items (Appendix B). For special tokens, $\gamma_h(t)$ is hard-pinned to 1 at the output (independent of the MLP value at $\bar{I}{=}0$; Appendices A and B). We use the pilot-selected amplification cap $s_{\max}{=}9$:

$$\gamma_h(t) = 1 + s_{\max}\ \sigma\big(\text{MLP}_{\text{IWVS}}(\bar{I}(t))\big)_h, \qquad t \notin \mathcal{S}. \tag{6}$$

For non-special tokens, $\gamma_h(t)$ is amplification-only ($1 < \gamma_h(t) < 1 + s_{\max}$) but not constrained to be monotone in rarity; rare-token amplification is empirical (Appendix I.3). Scaling $\mathbf{v}_{hj}$ by $\gamma_h(t_j)$ changes value contribution magnitude without changing attention weights.

Frequency-aware information gate (FAIG). FAIG adds a frequency-conditioned offset to IG gate logits at each query position. A query-side attention-logit offset would add the same constant across keys for a fixed query and cancel under row-wise softmax, while the post-attention IG gate is not row-normalized; FAIG therefore modulates how much aggregated context is written back:

$$\mathbf{g}_i = \sigma\left(\mathbf{x}_i W_{\text{IG}} + \beta \cdot \text{MLP}_{\text{FAIG}}(\bar{I}(t_i))\right) \tag{7}$$

where $\mathbf{g}_i \in [0,1]^{d_{\text{model}}}$ is the gate vector for position i and t_i is the query token at that position. The default elementwise variant uses $\text{MLP}_{\text{FAIG}} : \mathbb{R}^1 \to \mathbb{R}^{Hd_v}$ and a learnable layerwise scalar β. At $\beta{=}0$, FAIG reduces exactly to IG, so the gate-matched ablation isolates the query-token frequency pathway. A 20k-step pilot favors this additive offset over multiplicative and frequency-only gates (Appendix G, Table 18); variants and learned gate shifts are in Appendix I.1. All primary comparisons train from

scratch: FAB is neutral at step 0, but IWVS/FAIG and the IG backbone are active under standard linear initialization; Appendix D gives the exact initialization and Appendix I.3 reports empirical stability checks.

4 EXPERIMENTS

Setup. We pre-train LLaMA-style decoder-only Transformers (Touvron et al., 2023) with RoPE (Su et al., 2024), SwiGLU (Shazeer, 2020), and RMSNorm (Zhang & Sennrich, 2019) at 150M, 450M, and 1.2B parameters on FineWeb-Edu `sample/100BT` (Penedo et al., 2024). We use a local 25,024-token byte-level byte-pair encoding (BBPE) tokenizer (Sennrich et al., 2016; Wang et al., 2020); $P(t)$ is computed once during tokenizer/prior construction from the tokenized corpus and quantile-floor clamped (Appendix B). Language-model pre-training itself uses only the training shards, and held-out PPL evaluation uses the reserved final shard; Appendix N details the tokenizer, data splits, evaluation framework, and construction details for the fixed tokenizer-indexed unigram table. Training uses AdamW (Loshchilov & Hutter, 2019) with cosine decay for one epoch at context length 2048. We report held-out PPL/D0/decile PPL and zero-shot ARC-E/ARC-C (Clark et al., 2018), HellaSwag (HSwag) (Zellers et al., 2019), PIQA (Bisk et al., 2020), and LAMBADA (Paperno et al., 2016); 150M uses 3 seeds and 450M uses 2 seeds. Appendix D details the architecture and optimizer.

4.1 RESULTS

At the 150M scale, we first report the main Baseline vs. FAA comparison (mean ± standard deviation, 3 seeds; full statistics in Appendix L), then visualize the frequency-spectrum effect and decompose the components.

Table 1: **Main 150M results.** Mean ± standard deviation over 3 seeds; full seed statistics are in Appendix L.

Configuration	PPL↓	D0↓	ARC-E	ARC-C	HSwag	PIQA	LAMBADA
Baseline	11.36 ± 0.02	53.60 ± 0.14	56.96 ± 1.20	23.80 ± 0.66	32.14 ± 0.27	65.56 ± 0.28	31.71 ± 0.36
FAA (Ours)	**10.80 ± 0.01**	**48.85 ± 0.59**	**58.85 ± 0.54**	**25.86 ± 0.53**	**33.27 ± 0.05**	**66.18 ± 0.47**	**33.50 ± 0.49**
Δ	−4.9%	−8.9%	+1.89	+2.05	+1.13	+0.62	+1.79

Perplexity and decile PPL. We partition the non-special, non-single-byte vocabulary into ten frequency deciles by unigram probability (D0=rarest $10\%, \ldots,$ D9=most common 10%); each decile contains a similar number of vocabulary items but a highly skewed share of token occurrences (D0: 0.16%, D9: 52.91% of non-special occurrences). Decile boundaries are tokenizer-fixed and identical across Baseline and FAA, so any per-decile gap reflects loss differences rather than a change in decile membership. Table 1 shows a 4.9% reduction in overall PPL and an 8.9% reduction in D0 PPL (mean across 3 seeds). Single-byte tokens (the 256 raw bytes in BBPE) are excluded because raw-byte continuation predictions inside unmerged byte sequences can yield artificially low PPL that distorts decile diagnostics; the full breakdown is in Appendix M.

FAA reduces PPL across all frequency deciles, with larger gains for rare tokens than for common tokens (Appendix M, Figure 15). This pattern supports our hypothesis that injecting an extrinsic statistical prior into attention addresses a low-frequency learning bottleneck without sacrificing common-token modeling.

At 150M, FAA improves zero-shot accuracy on all five benchmarks (Table 1), with the largest gains on ARC-C (+2.05), ARC-E (+1.89), and LAMBADA (+1.79). Evaluation framework details are in Appendix N.

Component decomposition. Table 2 decomposes FAA into a shared gating backbone and frequency-specific pathways. The parameter overhead is dominated by the content-driven IG projec-

Table 2: **150M component ablation.** Single-run diagnostic; blue rows share the IG gate to isolate FAB+IWVS. *Sink* is mean attention to position 0; headline claims use multi-seed results.

Configuration	Params	PPL↓	D0↓	Sink	ARC-E	ARC-C	HSwag	PIQA	LAMBADA
Baseline	151.8M	11.39	53.79	0.0165	55.26	23.38	31.88	65.29	31.57
IG-only	158.9M	11.14	51.17	0.0026	57.24	25.09	32.70	66.49	32.06
FAB+IWVS	151.9M	11.04	49.90	0.0300	58.29	25.00	32.56	65.34	32.37
FAB+IWVS+IG	159.0M	10.82	48.15	0.0027	58.21	25.34	33.30	**67.14**	33.50
FAB+IWVS+FAIG (Ours)	159.3M	**10.79**	**48.02**	0.0027	**59.47**	25.77	**33.33**	65.67	**34.10**
FAB-only	151.8M	11.26	52.20	0.0226	56.52	24.91	32.10	65.67	31.79
IWVS-only	151.9M	11.48	54.75	0.0441	55.68	24.06	32.21	64.25	30.43
FAIG-only	159.2M	11.16	51.69	0.0026	55.43	24.32	32.49	65.34	32.37
FAB+IWVS+FAIG-Headwise	152.0M	11.02	49.88	0.0028	57.83	**25.94**	32.87	66.21	32.84
Δ (Ours − Baseline)	+4.9%	−5.3%	−10.7%	−0.0138	+4.21	+2.39	+1.45	+0.38	+2.53

tion, not by the corpus-frequency lookup or its small per-layer modules (Appendix C); hence the blue gate-matched rows are the key comparison for attributing gains.

Without any gate, FAB+IWVS outperforms IG-only on PPL (11.04 vs. 11.14) and D0 PPL (49.90 vs. 51.17), while adding only 0.043% parameters relative to Baseline, compared with 4.7% for IG-only. In the gate-matched comparison, adding FAB+IWVS to the same IG backbone improves PPL from 11.14 to 10.82 (−2.9%) and D0 from 51.17 to 48.15 (−5.9%), isolating a frequency-pathway gain beyond the shared gate. The full gate-matched decile breakdown (Appendix R) shows a larger relative gain for D0 (5.9%) than D9 (2.6%).

Building on the FAB+IWVS+IG backbone, FAIG completes the key/value/query decomposition by adding the only frequency pathway conditioned on the query-position token identity. With 0.3M additional parameters, FAIG improves 4/5 downstream benchmarks relative to FAB+IWVS+IG (largest gains: ARC-E +1.26, LAMBADA +0.60) and further reduces PPL (10.82 → 10.79) and D0 PPL (48.15 → 48.02).

Single-component rows further show IWVS alone underperforms Baseline (11.48 vs. 11.39), but its combination with FAB outperforms either component alone.

Mechanistically, the per-head output is $\sum_j A_{hij}\,\gamma_h(t_j)\mathbf{v}_{hj}$: IWVS can only amplify values that receive attention, whereas FAB can route more mass to high-self-information keys. Their combination therefore couples key-side routing with value-path amplification (we discuss the scope of this interpretation in Appendix G).

Comparison with frequency-aware baselines. To contextualize FAA against related long-tail and frequency-aware methods, we train 150M baselines *from scratch* under identical architecture, tokenizer, and training budget: a loss-level control (adapted focal loss (Lin et al., 2017)), training-time logit adjustment (Menon et al., 2021) at $\tau \in \{0.5, 1.0\}$, frequency-aware embedding bias, a frequency-aware language-model head (LM-head) bias motivated by prediction-head frequency analysis (Kobayashi et al., 2023), their combination, and the gate-free FAB+IWVS variant from Table 2.

Logit-adjustment baselines are evaluated with the training-matched bias, i.e., we report PPL of $\mathrm{softmax}(z + \tau \log P(t))$ rather than of the raw logits z, so the reported PPL is computed under the training-matched predictive distribution optimized by the loss.

Including FAB+IWVS here makes the comparison to embedding- and head-level interventions capacity-matched: FAB+IWVS has 151.9M parameters, close to the 151.8M gate-free baselines, yet outperforms those frequency-aware alternatives on PPL/D0. The full FAA model adds IG/FAIG on top, achieves the lowest PPL/D0, and improves all five downstream scores over FAB+IWVS (Appendix H.1, Table 19). Training-time logit adjustment yields modest improvements over Baseline (PPL 11.31 at τ=1.0, comparable to the LM-head bias) but does not reach FAA; stacking it with FAA does not add consistent gains (Appendix H.2, stacking analysis), suggesting that FAA already captures the bulk of the unigram-prior signal that output-logit-level adjustment can supply.

Table 3: **Design-choice controls at 150M.** Single-run diagnostics comparing parameter-matched and altered-prior variants.

Configuration	Params	PPL↓	D0↓	ΔPPL
Baseline (d_{model}=768)	151.8M	11.39	53.79	—
Wider Baseline (d_{model}=792)	160.2M	11.25	52.01	−1.2%
FAA + Random prior	159.3M	11.19	52.10	−1.8%
FAA + Shuffled prior	159.3M	11.13	50.96	−2.3%
FAA + Corpus prior	159.3M	**10.79**	**48.02**	−5.3%

4.2 Design-choice ablations

To verify that FAA's gains stem from the *aligned* corpus-derived frequency mapping rather than from added parameters or an arbitrary fixed per-token feature, Table 3 compares against a parameter-matched Wider Baseline (d_{model}=792), a Shuffled Prior (token↔ $I(t)$ permutation, marginal preserved), and a Random Prior (bootstrap-resampled $I(t)$, bijection broken). The Wider Baseline recovers only −1.2% PPL (FAA's PPL reduction is 4.3× larger). More importantly, the fixed unaligned Shuffled/Random priors closely match the IG-only gate baseline in Table 2 (11.13/11.19 vs. 11.14 PPL), whereas the aligned corpus prior reaches 10.79 PPL. Thus fixed unaligned prior-shaped features add little beyond gate capacity, while the aligned token-frequency mapping supplies the remaining gain.

Globally shared learned-token control. We replace the corpus-frequency table with a single learned vector $b \in \mathbb{R}^{|\mathcal{V}|}$, while keeping one token table shared across all layers and heads. FAB uses $\alpha_{\ell,h} b[t_j]$, and IWVS/FAIG consume the same b through fixed affine calibration; the content gate is unchanged. Corpus, midpoint, and shuffled initializations each use three seeds (Appendix Q). Table 4 reports overall PPL and all five downstream benchmarks. Across the reported means, static FAA and corpus initialization show a modest overall advantage over midpoint and shuffled initialization: both have slightly lower PPL and higher accuracy on most downstream tasks. Static FAA provides competitive PPL and generally favorable downstream performance without extra token-wise learned parameters, with higher mean accuracy than the corpus-initialized control on four of five tasks. Across-seed variation and control details are reported in Appendix Q. Corpus initialization also preserves strong frequency alignment while allowing token-specific adjustments (Pearson $r = 0.9819$–0.9827 across three seeds; Appendix Q).

Table 4: **Shared-token learnability at 150M.** Full FAA, averaged over three seeds; downstream accuracies are percentages. Extra token parameters: 0 for Static FAA, 25,024 for each learned row. Across-seed variation and control details are in Appendix Q.

Token table	PPL↓	ARC-E	ARC-C	HSwag	PIQA	LAMBADA
Static FAA	**10.80**	**58.85**	**25.86**	33.27	66.18	**33.50**
Corpus-init	**10.80**	58.78	24.72	**33.30**	65.94	33.42
Midpoint-init	10.81	58.68	24.80	33.25	**66.32**	33.18
Shuffled-init	10.81	58.31	25.00	33.05	65.80	32.96

Beyond these prior-alignment controls, 20k-step pilots support our choice of additive FAB and FAIG offsets (Appendix G, Table 18), PPL varies by at most 0.02 across quantile-floor percentiles (Appendix B, Table 6), and the pilot-selected IWVS amplification cap $s_{\max}$=9 outperforms $s_{\max}$=4 on PPL, D0 PPL, and four of five downstream tasks (Appendix I.3, Table 26). A bidirectional IWVS variant ($\gamma_{\min}$=0.1) does not improve PPL over amplification-only, while vocabulary-ID and character-entropy reassignments yield similar PPL but lower ARC-E and ARC-C accuracy (Appendix H.2, Table 20). These ablations support amplification-only value scaling and the corpus-aligned token–prior assignment.

4.3 SCALABILITY AND TRANSFER

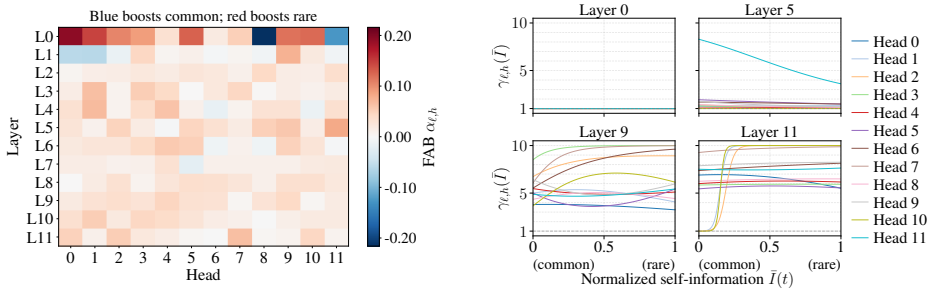
FAA lowers PPL from 9.29 to 8.98 at 450M and from 8.35 to 8.04 at 1.2B, with respective D0 PPL reductions of 6.2% and 6.4% (Appendix L.1, Table 30). Results are averaged over two seeds at 450M and reported for a single run at 1.2B; per-seed 450M results are provided in Appendix L.2 (Table 31).

At 450M, FAA improves all five benchmarks in both seeds (mean gains: ARC-E +2.27, ARC-C +2.73, LAMBADA +3.45); at 1.2B, FAA improves 4/5 benchmarks with largest gains on HSwag (+2.37) and LAMBADA (+4.13), alongside a small ARC-E regression (-0.46). PPL and D0 PPL reductions hold across all three scales, with multi-seed validation at 150M (3 seeds) and 450M (2 seeds) and a single run at 1.2B.

FAA also transfers across architectures and zero-shot domains (Appendix O): on a GPT-NeoX-style 150M with LayerNorm, GELU MLP, and parallel attention-MLP residuals (Biderman et al., 2023; Black et al., 2022), Full FAA reduces PPL and D0 PPL in all three seeds. On PubMed biomedical articles (Cohan et al., 2018) without recomputing target-domain priors, FAA still reduces PPL and D0 PPL.

PubMed continued pre-training (CPT) results are provided in Appendix P.

5 ANALYSIS



(a) FAB $\alpha_{\ell,h}$ values.

(b) Representative IWVS $\gamma_{\ell,h}(\bar{I})$ response.

Figure 2: **Interpretability snapshot at 150M.** (a) Most learned FAB $\alpha_{\ell,h}$ values are positive. (b) IWVS stays near identity in early layers, while many later-layer heads amplify rare, high-self-information tokens more strongly.

Averaged across three seeds, Baseline assigns more attention per key to the most-frequent decile (D9) than to the rarest decile (D0), whereas FAA reverses this ordering, favoring rare keys. Mean received attention reaches $2.41\times$ the Baseline level for D0, compared with $1.56\times$ for D9 (Table 5). We average attention received per key over causally eligible queries, layers, and heads, using 65,536 held-out token occurrences per checkpoint (Appendix S.1). An illustrative attention map for the *mitochondria* probe is provided in Appendix K (Figure 12).

Table 5: **FAA reverses the rare/frequent-key attention ordering.** Baseline assigns more attention per key to D9 (most frequent), whereas Full FAA assigns more to D0 (rarest). Three-seed mean $\pm$ population standard deviation; ratios are Full FAA/Baseline.

	D0 (rarest)	D9 (most frequent)
Baseline	0.00643±0.00015	0.00825±0.00123
Full FAA	0.01548±0.00006	0.01290±0.00018
Full FAA / Baseline	2.41×	1.56×

Learned FAA parameters reveal depth-specific specialization (Figure 2; full FAB values in Appendix I.2). FAB $\alpha_{\ell,h}$ has highest head-level variance at L0 and becomes predominantly positive in deeper layers, consistent with Artzy & Schwartz (2024), who describe a two-phase pattern in which lower layers gather information from previous-token representations and higher layers consolidate

that information. Quantitatively, 88.2% of FAB $\alpha_{\ell,h}$ values are positive, indicating a broad but not universal rare-key preference.

IWVS $\gamma_{\ell,h}(\bar{I})$ stays near unity early but reaches up to $10\times$ for many deep-layer heads at high self-information (Figure 11; stability in Appendix I.3). On a representative evaluation batch, mean scales rise from ≈ 1.0 in L0–L3 to ≈ 1.3–2.5 in L4–L8 and ≈ 6–7 in L9–L11 (Figure 10); despite the $10\times$ cap, all three 150M seeds train without divergence spikes.

FAIG gate shifts are mixed-sign by layer and decile, not uniformly rare-opening (Appendix I.1, Figure 7).

Attention sink. We report a threshold-free diagnostic, $\mathrm{Sink} = (LH|\mathcal{Q}|)^{-1} \sum_{(b,i)\in\mathcal{Q}} \sum_{\ell,h} A_{i,0}^{(b,\ell,h)}$, the mean post-softmax attention to key position 0 over layers, heads, and unmasked batch-query positions. This includes the forced $i{=}0$ causal term, so absolute values are convention-specific and gaps come from $i{\geq}1$ queries. Ablations show sink–quality decoupling: IWVS-only and gate-free FAB+IWVS raise sink (0.0441, 0.0300), yet FAB+IWVS improves PPL/D0 and all 5/5 downstream benchmarks; IG-only and final FAA have nearly identical sinks (0.0026 vs. 0.0027) despite different PPL/D0. Thus sink mitigation is primarily attributable to IG, whereas FAB+IWVS/FAIG are evaluated by rare-token PPL/D0 and downstream gains, consistent with conditional sink findings (Gu et al., 2025; Xiao et al., 2024; Yu et al., 2024).

Efficiency. FAA adds $+4.9\%$ parameters (7.5M, dominated by W_{IG}). Forward-pass benchmarks using standard matrix-multiplication/softmax attention for both the Baseline and Full FAA show $+5.4\%$ peak memory and a 7.0% throughput reduction (Appendix C, Tables 7 and 8). The dominant extra compute is one additional $d_{\text{model}} \times d_{\text{model}}$ projection per layer ($\approx$4.7% FLOPs at $d_{\text{model}}{=}768$, $T{=}2048$). FAA retrieves precomputed token corpus-frequency priors from a vocabulary-indexed table in $O(1)$ time. During cached decoding, IWVS scales each new value once and stores the scaled value in the KV cache; later queries reuse it without recomputing its scale or rescaling it. FAIG computes the current token's output gate from its hidden state and prior, without caching past gates. For optimizer-step and cached-decoding measurements, we use a FlashAttention-2 (Dao, 2024) extension that, like FlashBias (Wu et al., 2025), supports additive attention bias. The extension adds FAB's per-key bias inside the attention kernel and caches each new key's bias once for later queries. At 150M, token-ID/FAB metadata adds 0.792% to logical K/V storage; isolated FAB-only prior bookkeeping adds 0.28–0.57% decode latency on H100 (Appendix C.4). Appendix C.1 details the bias implementation. Full FAA's H100 optimizer-step throughput is 11.89%/10.22% below Baseline at 150M/1.2B (Appendix C.3, Table 10). Despite this per-step overhead, FAA reaches the two fixed loss targets using 1.65–2.42$\times$ and 1.28–1.45$\times$ fewer tokens, respectively, and achieves lower FLOP-matched loss across all three seeds (Appendix E). Appendix E.1 adds time-to-target projections using measured throughput.

6 Conclusion

At 150M, gate-free FAA (FAB+IWVS) reduces PPL by 3.1% and D0 PPL by 7.2% in a representative run, with only 0.043% additional parameters. Full FAA reduces PPL by 4.9% and D0 PPL by 8.9% on average across three seeds, with 4.9% additional parameters. Gate-matched ablations, prior controls, and 450M/1.2B results support corpus frequency as a useful architectural prior.

Limitations. The static unigram prior cannot model context-conditional surprisal, may overweight noisy rare strings, and misses BBPE-split multi-token concepts. Multi-seed evidence covers 150M (including the learned-token and NeoX controls) and 450M; 1.2B and the fixed-prior controls remain single-seed. Calibration, memorization/extraction risk, multilingual use, and language-model quality with 100k+ vocabularies remain untested (Appendix T).

AI USE STATEMENT

AI assistants were used for literature search and review, experimental design, code implementation and review, interpretation of results, and manuscript drafting and revision. The authors checked AI-assisted claims and citations against source papers and experimental records, and reviewed and tested the relevant code. The authors take responsibility for the accuracy and integrity of the paper and its supporting materials.

REPRODUCIBILITY STATEMENT

Appendix D specifies model configurations, optimization settings, and initialization. Appendix N describes the datasets, tokenizer construction, data splits, and downstream evaluation. Anonymized core implementation, representative configurations, and a runnable example are provided in the supplementary material.

REFERENCES

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.

Amit Ben Artzy and Roy Schwartz. Attend first, consolidate later: On the importance of attention in different LLM layers. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 177–184, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.10.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 23–29 Jul 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pp. 95–136, virtual+Dublin, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.bigscience-1.9.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *International Conference on Learning Representations*, 2020.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Second Conference on Language Modeling*, 2025.

Tyler A. Chang and Benjamin K. Bergen. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16, 2022. doi: 10.1162/tacl_a_00444.

Woojin Chung and Jeonghoon Kim. Exploiting vocabulary frequency imbalance in language model pre-training. In *Advances in Neural Information Processing Systems*, volume 38, pp. 137676–137706. Curran Associates, Inc., 2025.

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. *arXiv preprint arXiv:1803.05457*, 2018.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2097.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9268–9277, 2019. doi: 10.1109/CVPR.2019.00949.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024.

Franck Dernoncourt and Ji Young Lee. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 308–313, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30318–30332. Curran Associates, Inc., 2022.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: pure attention loses rank doubly exponentially with depth. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2793–2803. PMLR, 18–24 Jul 2021.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. FRAGE: Frequency-agnostic word representation. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations*, 2025.

Ankit Gupta and Jonathan Berant. Value-aware approximate attention. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9567–9574, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.753.

John Hale. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *International Conference on Learning Representations*, 2024.

Yanjin He, Qingkai Zeng, and Meng Jiang. Pre-trained models perform the best when token distributions follow Zipf's law. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28009–28021, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1421.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15696–15707. PMLR, 23–29 Jul 2023.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.574.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Transformer language models handle word frequency in prediction head. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4523–4535, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.276.

Martin Krallinger, Obdulia Rabal, Saber A. Akhondi, Martín Pérez Pérez, Jesús Santamaría, Gael Pérez Rodríguez, et al. Overview of the BioCreative VI chemical-protein interaction track. In *Proceedings of the Sixth BioCreative Challenge Evaluation Workshop*, volume 1, pp. 141–146, Bethesda, MD, USA, 2017.

Roger Levy. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177, 2008. doi: 10.1016/j.cognition.2007.05.006.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22947–22970. Curran Associates, Inc., 2024. doi: 10.52202/079017-0722.

Haoran Lian, Yizhe Xiong, Zijia Lin, Jianwei Niu, Shasha Mo, Hui Chen, Peng Liu, and Guiguang Ding. LBPE: Long-token-first tokenization to improve large language models. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, April 2025. doi: 10.1109/ICASSP49660.2025.10889733.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. doi: 10.1109/ICCV.2017.324.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 07–08 Apr 2022.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144.

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 *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849. Curran Associates, Inc., 2024. doi: 10.52202/079017-0970.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems*, volume 38, pp. 100092–100118. Curran Associates, Inc., 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Jason Ramapuram, Federico Danieli, Eeshan Gunesh Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, and Russell Webb. Theory, analysis, and best practices for sigmoid self-attention. In *International Conference on Learning Representations*, 2025.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 840–854, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.59.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162.

Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

Noam Shazeer. GLU variants improve Transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. doi: 10.1108/eb026526.

Alessandro Stolfo, Ben Wu, Wes Gurnee, Yonatan Belinkov, Xingyi Song, Mrinmaya Sachan, and Neel Nanda. Confidence regulation neurons in language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 125019–125049. Curran Associates, Inc., 2024. doi: 10.52202/079017-3970.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *First Conference on Language Modeling*, 2024.

Lintang Sutawika, Hailey Schoelkopf, Leo Gao, Baber Abbasi, Stella Biderman, Jonathan Tow, ben fattori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas, Thomas Wang, sdtblck, nopperl, gakada, tttyuntian, researcher2, Julen Etxaniz, Chris, Hanwool Albert Lee, Leonid Sinev, Zdeněk Kasner, Kiersten Stokes, Khalid, KonradSzafer, Jeffrey Hsu, and Anjor Kanekar. EleutherAI/lm-evaluation-harness: v0.4.9.1. Zenodo software release, doi: 10.5281/zenodo.16737642, August 2025.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008. Curran Associates, Inc., 2017.

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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. Neural machine translation with byte-level subwords. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9154–9160, 2020. doi: 10.1609/aaai.v34i05.6451.

Yifei Wang, Yueqi Wang, Zhenrui Yue, Huimin Zeng, Yong Wang, Ismini Lourentzou, Zhengzhong Tu, Xiangxiang Chu, and Julian McAuley. FASA: Frequency-aware sparse attention. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 932–948, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.72.

Haixu Wu, Minghao Guo, Yuezhou Ma, Yuanxu Sun, Jianmin Wang, Wojciech Matusik, and Mingsheng Long. FlashBias: Fast computation of attention with bias. In *Advances in Neural Information Processing Systems*, volume 38, pp. 20729–20756. Curran Associates, Inc., 2025. doi: 10.52202/085713-0699.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024.

Sangwon Yu, Jongyoon Song, Heeseung Kim, Seongmin Lee, Woo-Jong Ryu, and Sungroh Yoon. Rare tokens degenerate all tokens: Improving neural text generation via adaptive gradient gating for rare token embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29–45, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.3.

Zhongzhi Yu, Zheng Wang, Yonggan Fu, Huihong Shi, Khalid Shaikh, and Yingyan Celine Lin. Unveiling and harnessing hidden attention sinks: Enhancing large language models without training through attention calibration. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57659–57677. PMLR, 21–27 Jul 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488. Curran Associates, Inc., 2022.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *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.

Rodolfo Zevallos, Mireia Farrús, and Núria Bel. Frequency balanced datasets lead to better language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7859–7872, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.527.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12360–12371. Curran Associates, Inc., 2019.

George Kingsley Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley Press, Cambridge, Mass., 1949.

Adam Zweiger, Jyo Pari, Han Guo, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 74084–74115. Curran Associates, Inc., 2025.

A ALGORITHM DETAILS

Algorithm 1 presents the complete FAA forward pass for a single attention layer. Full FAA adds FAB, IWVS, and FAIG to the IG backbone inherited from Gated Attention (Qiu et al., 2025); MLP modules implement the learned scalar responses, quantile floor clamping supplies the prior floor, and RoPE is applied to the standard projections. Throughout, all logarithms are natural (information in nats). Note that unigram log-probabilities $\log P(t)$ are precomputed from corpus statistics and stored as an $O(|\mathcal{V}|)$ lookup table $\boldsymbol{\lambda}$, requiring only $O(BT)$ lookups ($O(T)$ per sequence) during training and inference. As in the body, the learned gate-logit bias is omitted from notation; Appendix C gives the exact fused QKV-gate bias accounting.

Algorithm 1 FAA forward pass for one attention layer.

Require: Input $X \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ (layer ℓ), token IDs $t \in \{0,\dots,|\mathcal{V}|-1\}^{B\times T}$, unigram log-prob table $\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{V}|}$, non-special-token extrema $I_{\min}, I_{\max}$ precomputed from corpus, special-token set $\mathcal{S}$

Ensure: Output $O \in \mathbb{R}^{B\times T\times d_{\text{model}}}$

$\lambda \leftarrow \boldsymbol{\lambda}[t]$ {$O(BT)$ lookup from precomputed, quantile-floor-clamped table}
$Q, K, V \leftarrow XW^Q, XW^K, XW^V$ {Standard projections; RoPE applied to Q, K (omitted for brevity)}
$I \leftarrow -\lambda$ {Self-information from log-probs}
$\bar{I} \leftarrow (I - I_{\min})/(I_{\max} - I_{\min})$ {Normalize using non-special-token extrema}
Set $I_{b,j} = 0$ and $\bar{I}_{b,j} = 0$ wherever $t_{b,j} \in \mathcal{S}$ {Assign fixed reference statistics to special tokens}
$\boldsymbol{\Gamma} \leftarrow 1+s_{\max}\cdot\sigma(\text{MLP}_{\text{IWVS}}(\bar{I}))$ {One MLP call; non-special scales satisfy $1 < \boldsymbol{\Gamma}_{b,j,h} < 1+s_{\max}$}
Set $\boldsymbol{\Gamma}_{b,j,:}=1$ wherever $t_{b,j} \in \mathcal{S}$ {No IWVS amplification for special tokens}
for each head $h = 0,\dots,H-1$ **do**
  $S_h \leftarrow Q_h K_h^\top/\sqrt{d_k}$ {Standard attention logits}
  $S_h \leftarrow S_h + \alpha_{\ell,h}\cdot I$ {FAB: add per-key bias (broadcast over queries)}
  $A_h \leftarrow \text{softmax}(\text{causal_mask}(S_h))$ {Masked softmax}
  $V'_h \leftarrow V_h \odot \boldsymbol{\Gamma}_{:,:,h}$ {IWVS: scale each key's value by $\gamma_{\ell,h}(t_j)$ (broadcast over d_v)}
  $Y_h \leftarrow A_h \cdot V'_h$ {Weighted aggregation}
end for
$Z \leftarrow \text{Concat}(Y_0,\dots,Y_{H-1})$ {Per-token attention output}
$G \leftarrow \sigma(XW_{\text{IG}} + \beta_\ell \cdot \text{MLP}_{\text{FAIG}}(\bar{I}))$ {FAIG: elementwise frequency-aware gate; special tokens use $\bar{I}$=0}
$O \leftarrow (Z \odot G)W^O$ {Gate then output projection}
return O

Normalization rationale. We normalize self-information values to the range $[0, 1]$ using global non-special-token corpus statistics. This affine scaling preserves token ordering by frequency while stabilizing the input range for the IWVS and FAIG MLPs; special tokens are then mapped to the fixed reference value $\bar{I}$=0.

Complexity analysis. FAA adds frequency-conditioned computations to standard attention and, when the output gate is enabled, inherits the IG content-gate projection:

- **FAB**: $O(BT)$ for prior lookup and $O(BHT)$ to form head-specific key biases, which are added within the existing $O(BHT^2d_k)$ score computation
- **IWVS**: $O(BTH^2)$ for the 2-layer scalar-to-H MLP (hidden width $32H$ at 150M; Appendix D) plus $O(BHTd_v) = O(BTd_{\text{model}})$ for value scaling
- **IG content gate**: $O(BTd^2_{\text{model}})$ for the projection XW_{IG} when gated attention is enabled
- **FAIG frequency offset**: for fixed MLP hidden width, $O(BTd_{\text{model}})$ for the elementwise MLP_{FAIG} or $O(BTH)$ for the headwise ablation; applying the gate costs $O(BTd_{\text{model}})$ in both cases

Standalone computations outside the attention kernel are linear in sequence length. FAB's bias is added within the existing $O(T^2)$ score computation, so FAA preserves attention's $O(T^2)$ asymptotic

complexity while adding only constant-factor overhead for fixed model width. The IG content-gate projection dominates the additional dense-projection computation in our configurations. Because the frequency-path inputs depend only on token identity, inference uses tokenizer-indexed prior lookups. Implementation details are in Appendix C.1. During cached decoding, FAB reuses stored key biases in our FlashAttention-2 extension, IWVS stores and reuses scaled values, and FAIG gates only the current token's attention output. Appendix C.4 gives the tensor dimensions and measured metadata cost.

B QUANTILE FLOOR CLAMPING DETAILS

Motivation. Some tokens in the learned vocabulary can have zero count under the final greedy tokenization of the prior-estimation corpus, yielding $P(t) = 0$ and $I(t) = \infty$. Laplace (add-one) smoothing avoids zeros by assigning $P(t) = \frac{C(t)+1}{N+|\mathcal{V}|}$, but it slightly perturbs observed frequencies and can produce extremely small probabilities for unseen tokens when N is large, leading to very large self-information values that can dominate the frequency pathways. We instead apply a corpus-derived quantile floor.

Quantile floor clamping. For non-special tokens, quantile floor clamping leaves observed tokens above a corpus-derived floor unchanged and assigns that floor to unobserved and extreme-tail tokens:

$$\log P(t) = \begin{cases} \log \frac{C(t)}{N} & \text{if } C(t) > 0 \text{ and } \log \frac{C(t)}{N} \geq Q_p \\ Q_p & \text{if } C(t) = 0 \\ Q_p & \text{if } C(t) > 0 \text{ and } \log \frac{C(t)}{N} < Q_p \end{cases} \tag{8}$$

where $C(t)$ is token count, N is total corpus tokens, and Q_p is the p-th percentile of observed non-special log-probabilities. We use $p = 0.5\%$ (log-probability floor ≈ -20.04, probability floor $\approx 2.0 \times 10^{-9}$), which affects only 303 tokens (1.2% of vocabulary, including all unobserved tokens and a small number of extreme-tail observed tokens). Tokens above the floor preserve their empirical frequencies exactly.

Special token handling. The five structural special tokens (IDs 0–4: BOS, EOS, PAD, UNK, and MASK) receive neutral self-information $I(t) = 0$. The 256 single-byte BBPE tokens (IDs 5–260) retain their corpus-derived frequencies; they are excluded only from the subword decile diagnostics, not from FAA's prior inputs. We also set $\bar{I}(t) = 0$ for the five structural special tokens before feeding the frequency pathways, ensuring:

- A zero additive FAB offset, so these tokens are not assigned artificial rarity
- No amplification in IWVS (we set the IWVS scale to 1.0 for special tokens)
- FAIG receives the fixed reference value $\bar{I}(t) = 0$ for these tokens. This removes any rare-token signal, but unlike IWVS there is no separate hard override that forces the resulting FAIG bias to zero. We do not hard-pin FAIG for special tokens because the gate is post-attention: a small additive offset on the gate logit cannot redirect attention or rescale value vectors, only modulate the post-aggregation residual write-back, so the asymmetric treatment is mechanistically motivated.

The $\bar{I}(t)$ override is separate from setting $I(t) = 0$. If min-max normalization were applied naively to $I(t) = 0$ using non-special-token extrema, it would yield $\bar{I}(t) = -I_{\min}/(I_{\max} - I_{\min}) < 0$, since $I_{\min} > 0$ for non-special natural-language tokens. We therefore compute $I_{\min}$ and $I_{\max}$ over non-special tokens and then explicitly map special tokens to $\bar{I} = 0$.

Probability mass. Quantile floor clamping modifies non-special-token log-probabilities without renormalization, so the implied non-special mass before special-token neutral overrides, $\sum_{t \notin \mathcal{S}} \exp(\log P(t))$, can exceed 1 by at most $\approx 6.0 \times 10^{-7}$ (303 clamped tokens at $\exp(-20.04) \approx 2.0 \times 10^{-9}$). This is inconsequential: the clamped values are consumed only as additive FAB biases or as inputs to min-max normalization for IWVS/FAIG, not as a sampling distribution. A global additive shift applied to all visible keys becomes a row-wise constant in FAB and cancels inside the softmax; it also cancels under min-max normalization. This algebraic invariance assumes the same transform for every compared key; the explicit special-token overrides remain a separate convention.

Percentile sensitivity. Table 6 shows FAA is robust to quantile-floor percentile choice, varying by at most 0.02 in PPL across settings.

Table 6: Quantile-floor ablation on 150M FAA. Performance is stable across percentile choices.

Percentile	Floor	PPL↓
$P_{1\%}$ (minimal)	−18.70	10.81
$P_{0.5\%}$ (default)	−20.04	**10.79**
$P_{0.1\%}$ (aggressive)	−21.62	10.80

C Parameters and computational efficiency

Table 7 details FAA parameter overhead by component.

Table 7: FAA parameter breakdown across scales.

Component	150M	450M	1.2B
FAB $\alpha_{\ell,h}$	144	384	512
IWVS MLP	64,656	221,568	557,568
IG	7,114,752	25,264,128	67,239,936
FAIG MLP + β_ℓ	304,908	812,568	1,082,384
Total FAA	7,484,460	26,298,648	68,880,400
Overhead	+4.9%	+5.8%	+5.9%

IG (elementwise gating) dominates FAA overhead (95–98% across scales). The frequency-specific components (FAB $\alpha_{\ell,h}$, IWVS MLP, and FAIG β_ℓ + MLP) are lightweight relative to the content gate—the cost of injecting the extrinsic frequency prior. The implementation appends the IG gate slice to the fused QKV projection for efficiency. When IG is enabled, that fused projection uses a bias vector for all Q, K, V, and gate slices, even though the ungated LLaMA-style Baseline has no QKV bias. The parameter table therefore includes the gate projection weights plus the fused bias terms; at 150M this accounts for the apparent extra $4d_{\text{model}}$ bias parameters per layer (of which $3d_{\text{model}}$ are Q/K/V bias slices). The comparisons Baseline vs. FAB+IWVS (both ungated, without QKV bias), IG-only vs. FAB+IWVS+IG (both gated, with QKV bias), and FAB+IWVS+IG vs. Full FAA (both gated, with QKV bias) hold this bias choice fixed.

C.1 Bias implementation

FAB has a separable bias: before causal masking, $B_{hij} = \alpha_h I(t_j)$ is constant across query positions and has rank at most one. This is an exact special case of the low-rank bias formulation of FlashBias (Wu et al., 2025), with a constant query-side factor.

Our FlashAttention-2 extension. We extend FlashAttention-2 (Dao, 2024) to accept the key bias $b_j = \alpha_h I(t_j)$ as a $(B \text{ or } 1, H, T)$ tensor while keeping Q/K/V at $D = 64$. With $s = 1/\sqrt{d_k}$, it adds b_j/s directly to the attention scores inside the kernel before the scaled online softmax, giving $s(QK^\top + b/s) = sQK^\top + b$. The backward pass accumulates bias gradients in FP32. The stored Q/K/V tensors require no auxiliary coordinate or padding.

Q/K augmentation. Alternatively, the same bias can be encoded exactly without singular value decomposition or learned approximation. Appending $q_{\text{bias}} = 1/s$ and $k_{\text{bias}} = b_j$ gives $s(\mathbf{q}_{hi}^\top \mathbf{k}_{hj} + q_{\text{bias}} k_{\text{bias}}) = s\mathbf{q}_{hi}^\top \mathbf{k}_{hj} + b_j$, exactly Eq. (3). This works with unmodified FlashAttention-2; at logical $D = 64$, our wrapper pads the augmented Q/K and the values to physical $D = 72$. Both implementations compute the same FAB; their costs below are measured on our models.

C.2 FORWARD-PASS OVERHEAD

To separate architectural overhead from attention-backend effects, the Baseline and Full FAA compute attention using matrix multiplication ($QK^\top$ and AV) and softmax, without FlashAttention, with `torch.compile(dynamic=True)` on one NVIDIA A100 80 GB graphics processing unit (GPU). We time forward passes under `torch.no_grad()`: 20 warmup iterations, then ten trials of 50 iterations, discarding the minimum and maximum trial throughputs. Tables 8 and 9 report the mean overhead and its batch/context breakdown. These measurements isolate full-context forward work without gradient computation or optimizer updates.

Table 8: Forward-pass overhead at 150M, averaged across nine batch/context settings using the same matrix-multiplication/softmax attention implementation for the Baseline and Full FAA.

Metric	**FAA Overhead**
Parameters	+4.9% (151.8M → 159.3M)
Throughput	−7.0% (range: 4–10%)
Peak Memory	+5.4% (range: 2–8%)

Table 9: Per-configuration forward-pass throughput (tokens/s) and peak-memory overhead across batch×context settings.

B×T	**Baseline**	**FAA**	**ΔThroughput**	**ΔMemory**
1×512	34.3K	31.0K	−9.6%	+8.3%
1×1024	41.4K	39.2K	−5.3%	+8.3%
1×2048	34.9K	33.1K	−5.0%	+6.3%
8×512	49.9K	46.1K	−7.7%	+5.3%
8×1024	46.7K	44.7K	−4.1%	+6.4%
8×2048	39.3K	35.9K	−8.7%	+3.6%
32×512	55.7K	52.7K	−5.4%	+2.4%
32×1024	50.3K	46.3K	−8.0%	+5.4%
32×2048	41.3K	37.5K	−9.1%	+3.0%
Average	—	—	−7.0%	+5.4%

C.3 OPTIMIZER-STEP THROUGHPUT

We measure complete optimizer steps on one H100 80GB with bfloat16 (BF16), $T = 2048$, and `torch.compile`. After five warmup steps, ten trials each time 50 optimizer steps on a fixed real batch; the minimum and maximum trials are discarded. Timing includes forward, backward, gradient clipping, AdamW, and zero-grad, but excludes data loading. The 150M batch is 32 and the 1.2B batch is 8.

Table 10 reports throughput and its change relative to a no-bias $D = 64$ Baseline. The Baseline is measured in the same session and uses PyTorch scaled dot-product attention (SDPA) with its FlashAttention backend. FAB-only isolates the score-bias path; Full FAA also includes IWVS and the IG/FAIG output gate. At 1.2B, for example, the extension reduces throughput by 1.71% for FAB-only and 10.22% for Full FAA, so the full cost cannot be attributed solely to the bias or padding. Section E.1 combines measured throughput with the tokens needed to reach fixed loss.

Table 10: Optimizer-step throughput (tokens/s), H100, BF16, $T = 2048$. Parentheses show throughput change (%) from the no-bias width-64 Baseline measured in the same session. FAA implementations are defined in Section C.1. Timing includes forward, backward, gradient clipping, AdamW, and zero-grad, but excludes data loading.

Scale / batch	FAA path	Baseline	FAB-only	Full FAA
150M (32)	Our extension	383,468	372,996 (−2.73%)	337,887 (−11.89%)
150M (32)	Q/K augmentation	383,468	333,509 (−13.03%)	308,560 (−19.53%)
1.2B (8)	Our extension	61,308	60,262 (−1.71%)	55,042 (−10.22%)
1.2B (8)	Q/K augmentation	61,278	57,333 (−6.44%)	52,568 (−14.21%)

C.4 KV-CACHE OPERATION AND BOOKKEEPING

For batch size B, L layers, H heads, cached length S, and head width d_v, each layer's cached values have shape $[B, H, S, d_v]$.

FAB stores shared token IDs $[B, S]$ and per-layer/head key biases $[L, B, H, S]$. Each new bias column is computed once at insertion. Subsequent queries read and add these cached biases to their attention scores without recomputing past key biases or rebuilding a dense query–key bias matrix.

IWVS scales each new $v_j \in \mathbb{R}^{B\times H\times 1\times d_v}$ by $[B, H, 1]$ coefficients, broadcast as $[B, H, 1, 1]$, and stores the scaled value $\tilde{v}_j = \gamma_h(t_j)v_j$ in the KV cache. Subsequent queries reuse $\tilde{V}$ without recomputing past scales or rescaling past values. The cache stores only the scaled values, not both original and scaled copies.

FAIG computes the current query-position token's gate from its hidden state and frequency prior, then applies it to the current attention output before the output projection. It neither rescales the KV cache nor stores past gates. The content projection is still computed for each new token; only the frequency-dependent response can use vocabulary-indexed lookup tables. Reading past K/V remains standard attention work, as in the Baseline.

At 150M, logical BF16 K/V storage is $2 \times 12 \times 12 \times 64 \times 2 = 36{,}864$ bytes per token. The extension adds an int32 token ID (4 bytes) and $12 \times 12 \times 2 = 288$ BF16 FAB-bias bytes: 292 bytes (0.792% of K/V), or 584 KiB alongside 72 MiB of K/V at $S = 2048$. Corpus-prior and vocabulary-indexed response tables are model-level inference state shared across sequences.

Table 11 isolates FAB-only bookkeeping. Both conditions use the same K/V and bias-aware attention. The off condition prefills biases for replayed forced tokens, removing token-ID writes/reads, the current-token prior gather, and bias-column construction/writes.

Table 11: Paired FAB-only decoding using our extension, with prior bookkeeping off/on (H100; prompt length 512, 256 forced-token steps; six alternating trials). Times are per-trial median CUDA-event times, reported as mean ± population standard deviation. Both conditions retain bias-aware attention.

Batch	Off (ms)	On (ms)	Paired delta (ms)	Overhead
1	0.6964 ± 0.0012	0.7004 ± 0.0014	+0.0040	+0.57%
8	0.9492 ± 0.0007	0.9518 ± 0.0010	+0.0026	+0.28%

D TRAINING SETUP DETAILS

We pre-train both Baseline and FAA for a single epoch on FineWeb-Edu (∼103.4B tokens under our tokenizer) with context length 2048. Across scales (150M, 450M, and 1.2B; Table 12), we use global batch sizes of 0.52M, 0.26M, and 0.13M tokens per step (256, 128, and 64 sequences on 8 GPUs). Optimization uses AdamW with β_1=0.9, β_2=0.95, weight decay 0.1, and gradient clipping at 1.0. AdamW parameter groups apply weight decay only to matrix weights: all 1D parameters (biases and normalization parameters) and FAA frequency-control parameters (`fab_alpha`, `faig_alpha`, `faig_*_bias_proj`, and `iwvs_fc*`) use zero weight decay. We use a cosine learning rate decay

Table 12: Model configurations and parameter counts; IG accounts for most FAA overhead.

Scale	Layers	Heads	d_{model}	Params	
				Base	FAA
150M	12	12	768	151.8M	159.3M
450M	24	16	1024	454.1M	480.4M
1.2B	16	32	2048	1176.6M	1245.5M

schedule with 6.3% linear warmup, a minimum learning rate of 0.1× peak, and peak learning rates of 6×10^{-4} (150M), 4×10^{-4} (450M), and 3×10^{-4} (1.2B). Unless noted otherwise, experiments run on 8× NVIDIA A100 80 GB GPUs with `bfloat16` precision and `torch.compile`. We report three random seeds at 150M (Appendix L) and two random seeds at 450M (Appendix L.2); 1.2B and most controls are single runs unless explicitly marked.

Table 13 reports approximate active training budgets per run. The ranges reflect observed variation across completed 150M runs and scaled estimates for larger runs under the same 8× NVIDIA A100 80 GB setup; they exclude queue wait time, dataset download/preprocessing, benchmark evaluation, and checkpoint upload/download time.

Table 13: Approximate active compute budget per run. GPU-hours are wall-clock hours multiplied by 8 GPUs.

Stage	Configured steps	Wall-clock/run	GPU-hours/run
150M pre-training	197,175	1–2 days	200–350
450M pre-training	394,349	3–5 days	600–1,000
1.2B pre-training	788,574	8–14 days	1,500–2,700
150M PubMed CPT	50,000	6–12 hours	50–100

FAA hyperparameters and MLP architectures. For reproducibility, we specify the exact frequency-path parameterizations used in our 150M experiments:

- **IWVS**: $s_{\max}$=9. At each layer, MLP_{IWVS} is a 2-layer MLP with GELU: $\mathbb{R}^1 \to \mathbb{R}^{32H} \to \mathbb{R}^H$ (i.e., $\text{Linear}(1, 32H) \to \text{GELU} \to \text{Linear}(32H, H)$), producing one logit per head through a shared nonlinear scalar-to-head basis.
- **FAIG (elementwise; default)**: β_ℓ is a learnable scalar per layer. MLP_{FAIG} is a 2-layer MLP with GELU: $\mathbb{R}^1 \to \mathbb{R}^{32} \to \mathbb{R}^{Hd_v}$ ($\text{Linear}(1, 32) \to \text{GELU} \to \text{Linear}(32, Hd_v)$; with Hd_v=d_{model}), producing an elementwise bias term that matches the gate-logit dimensionality.
- **FAIG-Headwise (ablation)**: $\beta_{\ell,h}$ is learnable per head per layer. MLP_{FAIG} is $\mathbb{R}^1 \to \mathbb{R}^{16} \to \mathbb{R}^H$ ($\text{Linear}(1, 16) \to \text{GELU} \to \text{Linear}(16, H)$); the resulting per-head bias is broadcast across each head's d_v gate dimensions.

In all cases, $\bar{I}(t)$ is min-max normalized using corpus-level statistics computed over non-special tokens, and special tokens are explicitly mapped to $\bar{I}(t)$=0.

FAA initialization. From-scratch training runs use GPT-NeoX-style small base initialization for embeddings and non-output-projection linear weights, with standard deviation $\sqrt{2/(5d_{\text{model}})}$ (Black et al., 2022). Our training code additionally applies a residual-projection scale $1/(\sqrt{d_{\text{model}}}L)$ to attention output projections. MLP projections use the same base linear initialization rather than a separate residual-projection override. Only the FAB bias path is neutral relative to standard attention: FAB starts with $\alpha_{\ell,h}$=0, making Eq. (3) standard attention before learning. The gate/value paths are intentionally active at step 0 in the from-scratch setting, so FAA is not initialized to an exact Baseline computation. The IG projection follows the same base linear initialization as other non-attention-projection linear modules, and β_ℓ (or $\beta_{\ell,h}$ in the headwise variant) starts at 1.0. For IWVS, the final layer of MLP_{IWVS} has zero bias and randomly initialized weights under this initialization, so non-special token scales are centered near $1 + s_{\max}/2 = 5.5$ when $s_{\max}$=9; special-token scales remain

fixed at 1. The final layer of MLP_{FAIG} is also randomly initialized, so the frequency-conditioned gate offset can be nonzero at step 0. Appendix I.3 checks that this non-identity initialization does not cause training divergence in the 150M multi-seed runs.

E TRAINING DYNAMICS

Figure 3 compares training loss over 103.4B tokens across three random seeds. FAA consistently achieves lower loss than Baseline in all seeds: final loss 2.389 ± 0.008 vs. 2.440 ± 0.008.

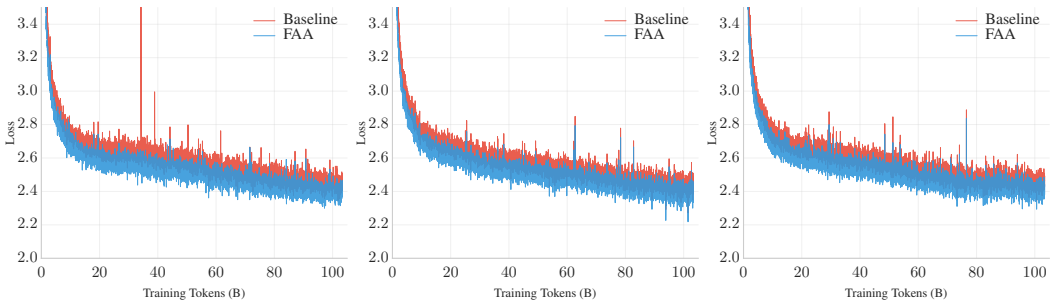


Figure 3: Training loss over 103.4B tokens across 3 seeds (150M). The y-axis clips one Baseline spike near 34B tokens.

Data efficiency. FAA requires fewer tokens to reach equivalent loss:

- 1.65–2.42× fewer tokens to reach loss 2.6 (17.5B–21.7B vs. 35.8B–42.4B)
- 1.28–1.45× fewer tokens to reach loss 2.5 (50.1B–56.5B vs. 69.6B–75.7B)

Since per-step training loss is noisy, tokens-to-loss is computed per seed using a trailing 100-point moving average of the logged loss, with linear interpolation between consecutive smoothed values; we report ranges across seeds. Under standard Transformer FLOPs accounting (fixed architecture and context length), fewer training tokens translate directly into lower training compute to reach a target loss; FAA's additional per-token FLOPs do not erase the observed token-to-target reductions. Wall-clock cost also depends on the attention implementation; Appendix E.1 combines token-to-target measurements with measured throughput, explicitly as a projection.

Optimization trace. The training curves show no divergence in the FAA runs across the three 150M seeds. We treat transient Baseline loss spikes in Figure 3 as descriptive only and do not claim a systematic stability advantage without a controlled optimization study.

Compute-matched comparison. We match *training compute* by total FLOPs. FAA's dominant additional FLOPs come from the IG projection (XW_{IG}), which adds one $d_{\text{model}} \times d_{\text{model}}$ linear per layer and costs $\approx 2d_{\text{model}}^2$ FLOPs per token (forward). With SwiGLU intermediate width $4d_{\text{model}}$, Baseline per-layer FLOPs per token are dominated by projections/MLP ($\approx 32d_{\text{model}}^2$) and attention matrix multiplications ($\approx 4d_{\text{model}}T$), so IG increases forward per-token FLOPs by $\frac{2d_{\text{model}}^2}{32d_{\text{model}}^2+4d_{\text{model}}T} \approx 4.7\%$ at d_{model}=768, T=2048. Since both Baseline and FAA are dominated by matrix multiplications, the forward-to-training multiplier (forward+backward) is approximately constant and the *relative* overhead is preserved; we therefore use this ratio to match total training FLOPs. Thus equivalent compute corresponds to 50B (FAA) vs. 52.3B (Baseline). Reported losses are local averages (±500 steps) over three seeds. FAA achieves 2.51 vs. Baseline's 2.56 (lower is better), and wins in all three seeds.

E.1 PROJECTED TIME TO FIXED TRAINING LOSS

The 150M curves in Figure 3 give paired tokens-to-target estimates using trailing 100-point moving averages and interpolation. We combine them with measured Full FAA/Baseline throughput ratios,

0.8811 for our extension and 0.8047 for Q/K augmentation:

$$\frac{\widehat{t}_{\mathrm{FAA}}}{\widehat{t}_{\mathrm{Baseline}}} = \frac{N_{\mathrm{FAA}}/N_{\mathrm{Baseline}}}{\mathrm{throughput}_{\mathrm{FAA}}/\mathrm{throughput}_{\mathrm{Baseline}}}.$$

Table 14 reports ranges of paired-seed projections, not ratios formed by arbitrarily combining endpoints from different seeds. They do not include data-loading or checkpoint overhead, do not replace a direct wall-clock run, and are not extrapolated to the 1.2B model.

Table 14: Projected FAA/Baseline time to fixed training loss at 150M for the two implementations in Table 10. Ratios combine paired three-seed token requirements with measured H100 throughput; they are not direct wall-clock training measurements.

Loss	Baseline / FAA tokens	Token ratio	Our extension	Q/K augmentation
2.6	35.8–42.4B / 17.5–21.7B	1.65–2.42×	0.47–0.69×	0.51–0.75×
2.5	69.6–75.7B / 50.1–56.5B	1.28–1.45×	0.79–0.89×	0.86–0.97×

F ATTENTION SINK ANALYSIS

Sink metric. Xiao et al. (2024) identify attention sinks as disproportionate attention to initial tokens, and Gu et al. (2025) define $\mathrm{Sink}_1^{\epsilon}$ as the fraction of heads across all layers whose mean first-token attention exceeds ϵ. We report a complementary threshold-free continuous diagnostic: the mean attention probability allocated to key position 0,

$$\mathrm{Sink} = \frac{1}{LH|\mathcal{Q}|} \sum_{(b,i)\in\mathcal{Q}} \sum_{\ell=0}^{L-1} \sum_{h=0}^{H-1} A_{i,0}^{(b,\ell,h)}, \tag{9}$$

where $A_{i,j}^{(b,\ell,h)}$ is the post-softmax attention weight at layer ℓ, head h, query position i, key position j for batch item b, and $\mathcal{Q}$ is the set of unmasked batch-query positions in a representative evaluation batch sampled from the held-out FineWeb-Edu evaluation shard. At $i{=}0$, causal masking forces $A_{0,0}^{(b,\ell,h)}{=}1$ for every model; the reported values include this common term, so absolute sink values should be interpreted under this convention, while differences across configurations are driven by positions $i \geq 1$. Quantitatively, the $i{=}0$ contribution adds $1/T \approx 5 \times 10^{-4}$ at our context length $T{=}2048$ and is identical across configurations, well below FAA's observed sink of 0.0027, so all reported gaps reflect $i \geq 1$ allocation rather than the masked diagonal. This continuous metric tracks the categorical attention-sink phenomenon (Gu et al., 2025; Xiao et al., 2024) without requiring a threshold.

Attribution and sink-quality decoupling. The IG output gate is inherited from the Gated Attention module of Qiu et al. (2025), and is the main observed contributor to FAA's sink reduction in our 150M decomposition. Table 2 (representative run) shows IG-only already reduces sink from 0.0165 to 0.0026, consistent with the sink-reduction effect reported by Qiu et al. (2025). Our final FAA configuration attains 0.0027 (multi-seed mean), and FAIG vs. IG yields a negligible sink change (0%).

We use sink only as a behavioral diagnostic, not as a quality metric. Three representative-run comparisons make the decoupling explicit; PPL and D0 PPL are from Table 2, and sink values are measured on the same run (single-seed; sink in absolute units):

- FAB+IWVS (no gate) raises sink to 0.0300 (above the 0.0165 Baseline) yet improves PPL/D0 over IG-only (11.04/49.90 vs. 11.14/51.17).
- IWVS-only has the largest sink in the ablation (0.0441, ≈167% above Baseline) but its PPL is 11.48, comparable to Baseline (11.39).
- IG-only and FAIG-only have nearly identical sinks (0.0026 vs. 0.0026); the gate-matched comparison is decided by PPL/D0/downstream, not by sink.

We therefore treat sink mitigation as primarily explained by IG in the 150M ablation, and treat rare-token PPL/D0 reductions as the substantive evaluation of FAB+IWVS. This is consistent with prior work showing that sink effects are conditional rather than monotone quality indicators: initial sinks can support streaming stability (Xiao et al., 2024), hidden/later sinks are not uniformly helpful (Yu et al., 2024), and softmax coupling can induce key-bias-like behavior independent of value computation (Gu et al., 2025). More broadly, sigmoid self-attention has been analyzed as an alternative to softmax attention that changes normalization and coupling behavior (Ramapuram et al., 2025).

Per-head sink tables. Tables 15 and 16 report per-layer, per-head sink values for the representative run under Baseline and FAA. Averaging over layers and heads recovers the single-seed sink 0.0165→0.0027; the multi-seed mean ± standard deviation is (0.0185 ± 0.0017) → (0.0027 ± 0.0001).

Table 15: **Baseline per-head sink attention.** Representative 150M run; values report mean attention mass assigned to position 0.

	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11
L0	0.003	0.003	0.004	0.003	0.004	0.003	0.003	0.003	0.002	0.003	0.003	0.003
L1	0.001	0.003	0.003	0.002	0.003	0.003	0.002	0.002	0.003	0.002	0.002	0.003
L2	0.006	0.003	0.007	0.008	0.008	0.008	0.005	0.008	0.009	0.006	0.008	0.006
L3	0.020	0.021	0.023	0.022	0.022	0.021	0.021	0.018	0.018	0.017	0.022	0.019
L4	0.013	0.018	0.017	0.019	0.017	0.016	0.018	0.018	0.017	0.013	0.028	0.015
L5	0.009	0.013	0.015	0.018	0.016	0.017	0.016	0.014	0.014	0.014	0.023	0.014
L6	0.012	0.014	0.011	0.010	0.011	0.012	0.012	0.011	0.012	0.010	0.014	0.012
L7	0.013	0.008	0.012	0.009	0.010	0.010	0.010	0.013	0.008	0.009	0.011	0.012
L8	0.012	0.008	0.011	0.010	0.005	0.003	0.009	0.007	0.006	0.008	0.009	0.010
L9	0.037	0.030	0.035	0.022	0.027	0.023	0.035	0.029	0.035	0.030	0.032	0.045
L10	0.030	0.040	0.031	0.035	0.032	0.025	0.040	0.043	0.038	0.038	0.041	0.039
L11	0.034	0.053	0.045	0.035	0.028	0.030	0.029	0.026	0.031	0.025	0.037	0.044

Table 16: **FAA per-head sink attention.** Representative 150M run; values report mean attention mass assigned to position 0.

	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11
L0	0.002	0.002	0.002	0.001	0.001	0.002	0.002	0.003	0.003	0.004	0.002	0.002
L1	0.002	0.002	0.002	0.001	0.002	0.002	0.001	0.002	0.002	0.003	0.002	0.001
L2	0.002	0.001	0.002	0.001	0.002	0.002	0.002	0.001	0.004	0.002	0.001	0.002
L3	0.002	0.002	0.002	0.002	0.002	0.004	0.002	0.002	0.002	0.002	0.004	0.002
L4	0.001	0.003	0.004	0.003	0.002	0.001	0.004	0.001	0.003	0.001	0.006	0.003
L5	0.002	0.005	0.003	0.002	0.002	0.004	0.002	0.002	0.002	0.003	0.001	0.004
L6	0.002	0.004	0.002	0.002	0.002	0.002	0.004	0.001	0.011	0.003	0.002	0.006
L7	0.004	0.008	0.001	0.003	0.004	0.013	0.006	0.003	0.002	0.003	0.002	0.004
L8	0.004	0.002	0.004	0.002	0.003	0.004	0.002	0.004	0.003	0.003	0.003	0.001
L9	0.001	0.004	0.004	0.003	0.003	0.003	0.004	0.003	0.003	0.003	0.004	0.002
L10	0.003	0.003	0.001	0.003	0.007	0.002	0.001	0.001	0.001	0.001	0.003	0.002
L11	0.003	0.005	0.006	0.001	0.001	0.001	0.001	0.003	0.001	0.001	0.005	0.001

Layerwise sink reduction. Table 17 reports the per-layer sink for the same representative run. The Baseline sink is at most 0.0069 for L0–L2, then grows in deeper layers and peaks at 0.0361 (L10), whereas FAA stays small across all layers (peak 0.0044 at L7). The largest reductions therefore occur in late layers, where Baseline sinks are largest—again consistent with IG suppressing the deeper-layer sink growth that softmax-coupled attention develops without gating.

Table 17: **Per-layer sink attention comparison.** Representative 150M run; values report mean attention mass assigned to position 0, with reduction relative to Baseline.

Layer	Baseline	FAA	Reduction
L0	0.0030	0.0020	33.3%
L1	0.0025	0.0018	28.0%
L2	0.0069	0.0017	75.4%
L3	0.0204	0.0023	88.7%
L4	0.0173	0.0028	83.8%
L5	0.0152	0.0027	82.2%
L6	0.0117	0.0034	70.9%
L7	0.0103	0.0044	57.3%
L8	0.0082	0.0028	65.9%
L9	0.0317	0.0030	90.5%
L10	0.0361	0.0023	93.6%
L11	0.0348	0.0025	92.8%

Scale-dependent sink behavior. At larger scales, the late-layer sink growth becomes more pronounced. Across two seeds at 450M (24 layers), the mean Baseline sink increases from 0.018 in early layers to 0.053 in L18–L23, a 2.9× increase, while FAA reduces overall sink by 89.6% (Figure 4). At 1.2B (16 layers), FAA reduces sink by 84.8% across all layers (Figure 5). At the 150M scale where we have an IG-only ablation (Table 2), IG-only nearly matches the sink reduction of the final FAA model; larger-scale attribution among IG, FAB, and IWVS remains an open check. Because the increasing sink magnitude appears in the Baseline without an output gate, it reflects a depth-related effect rather than something specific to FAA.

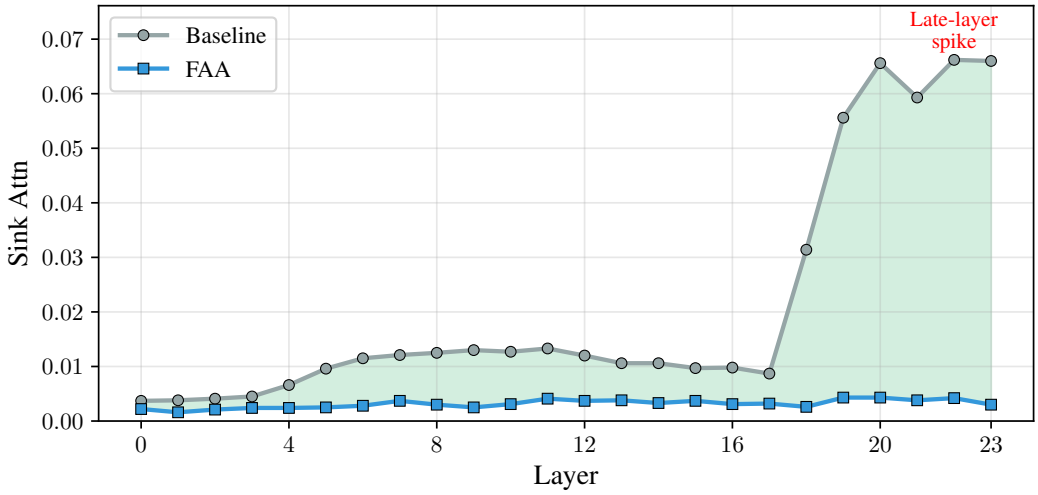


Figure 4: Representative-run per-layer sink at 450M. Baseline grows in late layers (L18–L23); FAA stays small across depth.

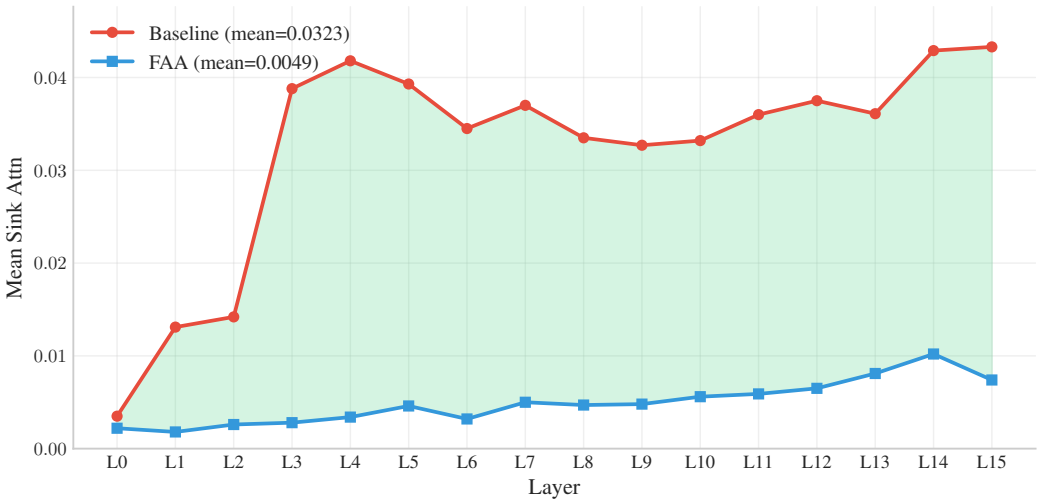


Figure 5: Per-layer sink at 1.2B. Baseline rises sharply at L3 and stays elevated; FAA stays small across all 16 layers.

G Design rationale and pilot ablations

Design rationale. Three choices are made intentionally:

- **Key/value/query asymmetry.** FAB is key-side (biasing *which* tokens are retrieved), IWVS is value-side (changing *how strongly* retrieved evidence contributes), and FAIG is query-side (controlling *how much* of the aggregated context is written back at position i).
- **Raw $I(t)$ for FAB, normalized $\bar{I}(t)$ for IWVS/FAIG.** When an affine transform is applied uniformly to all visible keys, a nonzero multiplicative factor can be absorbed into the learned α_h, and an additive shift cancels under row-wise softmax. The special-token overrides in Appendix B are applied separately. FAB therefore does not require a normalized input. IWVS and FAIG pass self-information through sigmoid nonlinearities that are scale-sensitive, so they consume $\bar{I}(t)$ to stabilize the MLP input range.
- **Min-max over corpus-level statistics, additive gate offsets.** Min-max normalization reflects that $I(t)$ is a global per-vocabulary-item statistic (defined from corpus-level counts), not a sequence-level distribution; pilot ablations (Table 18) show alternative normalizations yield equivalent PPL. Additive FAIG preserves $\mathbf{x}_i W_{\text{IG}}$ as the baseline computation and treats frequency as an offset rather than a replacement signal.

FAB multiplicative view. Eq. (4) shows that, for unmasked key positions, FAB reweights the content-based unnormalized attention weight by a learnable power of the key's unigram probability; masked positions retain zero attention probability. Unlike inverse document frequency (IDF) (Spärck Jones, 1972), which uses document counts, FAA's self-information uses token-occurrence counts. Structurally, FAB applies the additive form of logit adjustment (Menon et al., 2021) inside attention rather than at the output classifier: where training-time logit adjustment shifts class scores by $\tau \log P(c)$ with a fixed coefficient τ, FAB shifts attention scores over key positions by $-\alpha_{\ell,h} \log P(t_j)$ with a learned per-layer-head coefficient. Like ALiBi (Press et al., 2022), FAB injects an additive attention-logit bias outside the query–key dot product; unlike ALiBi's fixed distance penalty, FAB uses a learned coefficient on the key token's corpus frequency. Prior analyses of neural machine translation (NMT) encoders find frequency-sensitive attention heads that often attend to rare source tokens (Voita et al., 2019); we use this only as qualitative motivation because our setting is decoder-only causal self-attention.

Why amplification-only IWVS. SIF-style sentence embeddings motivate aggregation-side frequency weighting, but IWVS is not intended to copy SIF's fixed downweighting rule. In the attention

Table 18: **20k-step design-choice pilots at 150M.** Each group varies one design axis; the 20k Baseline is 13.79 PPL / 80.70 D0.

Design axis	Variant	PPL↓	D0↓
FAB form	Additive (ours)	12.98	**71.50**
	Multiplicative	**12.97**	71.65
Normalization	Min-max (ours)	**12.98**	**71.50**
	Midpoint/quarter-range	**12.98**	71.87
	None (raw)	**12.98**	71.84
FAIG gate	Additive (ours)	**12.98**	**71.50**
	Multiplicative	13.22	74.41
	Frequency-only (no content)	13.32	74.77

block, attenuation is already available through two mechanisms: the row-normalized attention distribution can route little mass to a key, and the IG/FAIG gate can suppress the post-aggregation write-back with $G \in (0,1)^{d_{\text{model}}}$. Pre-attention RMSNorm also normalizes each block input. We therefore assign IWVS the narrower role of selective value amplification for tokens that are both retrieved by attention and assigned high learned value-path scale. This design choice is empirical but not arbitrary: the bidirectional IWVS variant in Table 20 permits downscaling below identity and does not improve PPL over amplification-only.

Interpretation scope. The FAB–IWVS complementarity discussed in the body follows directly from the attention output $\sum_j A_{hij}\gamma_h(t_j)\mathbf{v}_{hj}$: FAB changes routing weights, whereas IWVS rescales selected values. We use this as a mechanistic interpretation of why the two pathways combine well, not as a causal mediation estimate.

Pilot ablations (20k steps). We use 20k-step pilot runs only for implementation selection before launching the full 150M/450M/1.2B experiments. Table 18 summarizes the design choices referenced in Section 4.2: additive FAB and min-max normalization perform on par with more complex alternatives, while an additive FAIG bias outperforms multiplicative and frequency-only gates. By "multiplicative FAB" we mean the variant $s_{hij} = (\mathbf{q}_{hi}^\top \mathbf{k}_{hj}/\sqrt{d_k}) \cdot (1 + \alpha_h \bar{I}(t_j))$, which scales rather than shifts the pre-softmax score (distinct from the multiplicative re-expression of additive FAB in Eq. (4)); "multiplicative FAIG" means the two-gate variant $\mathbf{g}_i = \sigma(\mathbf{x}_i W_{\text{IG}}) \odot \sigma\big(\beta \cdot \text{MLP}_{\text{FAIG}}(\bar{I}(t_i))\big)$, rather than the additive gate-logit offset in Eq. (7).

H Additional baselines and prior controls

H.1 Direct baseline comparisons

We train 150M baselines *from scratch* under the same architecture, tokenizer, and training budget as FAA: output-logit-level **training-time logit adjustment** (Menon et al., 2021) (adding $\tau \log P(t)$ to logits before cross-entropy); a frequency-aware **LM-head bias** motivated by prediction-head frequency analysis (Kobayashi et al., 2023); a frequency-aware **embedding bias** at the input; their **combination** (frequency-aware embedding + LM-head); and a loss-level control, adapted **focal loss** (Lin et al., 2017). For logit adjustment we report $\tau \in \{0.5, 1.0\}$, and for focal loss we use the same settings as the 150M configurations selected before the final comparison. For logit-adjustment baselines, we compute PPL from $\text{softmax}(z + \tau \log P(t))$, the same distribution used in the training loss.

For clarity, the non-FAA frequency baselines use the following exact intervention points. The frequency-aware embedding baseline adds a learned input-side offset,

$$\mathbf{x}_i = \text{Emb}(t_i) + \text{MLP}_{\text{emb}}(\bar{I}(t_i)), \qquad \text{MLP}_{\text{emb}} : \mathbb{R}^1 \to \mathbb{R}^{32} \to \mathbb{R}^{d_{\text{model}}}, \tag{10}$$

with GELU nonlinearity and otherwise standard attention. The LM-head bias baseline adds a learned-strength class prior after the output projection,

$$z'_{i,c} = z_{i,c} + \eta \log P(c), \tag{11}$$

Table 19: **Long-tail and frequency-aware baselines at 150M.** Same architecture, tokenizer, and budget; logit-adjustment PPL uses the training-matched $\mathrm{softmax}(z + \tau \log P(t))$ bias.

Configuration	PPL↓	D0↓	ARC-E	ARC-C	HSwag	PIQA	LAMBADA
Baseline	11.39	53.79	55.26	23.38	31.88	65.29	31.57
Adapted focal loss (Lin et al., 2017) (γ=2; hard-token control)	11.61	53.27	57.28	22.61	32.16	64.42	31.24
Training-time logit adjustment (Menon et al., 2021) (τ=0.5)	11.35	52.84	55.89	23.46	32.04	64.80	30.95
Training-time logit adjustment (Menon et al., 2021) (τ=1.0)	11.31	52.27	57.95	23.29	32.05	64.96	29.65
Frequency-aware embedding	11.22	51.48	57.53	22.10	32.17	65.29	31.46
Frequency-aware LM-head bias†	11.31	52.11	57.20	23.72	32.43	64.42	30.70
Frequency-aware embedding + LM-head†	11.27	52.25	58.54	23.38	32.12	64.80	30.12
FAB+IWVS (gate-free FAA)	11.04	49.90	58.29	25.00	32.56	65.34	32.37
FAA (Ours)	**10.79**	**48.02**	**59.47**	**25.77**	**33.33**	**65.67**	**34.10**

† LM-head bias is our implementation, motivated by prediction-head frequency analysis (Kobayashi et al., 2023).

where η is a learnable scalar initialized to 1.0; padded dummy vocabulary rows use the clamped rare-token floor in the softmax denominator. Training-time logit adjustment uses the same additive form with fixed τ inside the loss, $z'_{i,c} = z_{i,c} + \tau \log P(c)$. The adapted focal-loss baseline uses

$$\mathcal{L}_{\text{focal}}(i) = -(1 - p_{i,y_i})^{\gamma} \log p_{i,y_i}, \qquad p_{i,y_i} = \mathrm{softmax}(\mathbf{z}_i)_{y_i}, \quad \gamma = 2. \tag{12}$$

Table 19 shows that embedding- or LM-head-level frequency injection yields modest PPL reductions (best: -1.5% / -4.3% D0); the gate-free FAB+IWVS row gives a capacity-matched comparison against these gate-free baselines and improves further; the full FAA model adds the IG/FAIG gate and achieves the best PPL/D0 and downstream scores in this table. Focal loss gives a small D0 improvement (-1.0%) with a total-PPL trade-off (+1.9%), consistent with its hard-token focus being orthogonal to frequency. Training-time logit adjustment gives a small PPL reduction at both adjustment strengths (PPL 11.35 at τ=0.5, 11.31 at τ=1.0; D0 52.84 / 52.27; LAMBADA 30.95% / 29.65%): comparable to the LM-head bias on PPL, slightly worse than the embedding bias, and with no material difference between τ=0.5 and τ=1.0.

H.2 Design-choice ablations and prior controls

To test whether FAA's gains depend on the aligned corpus-derived frequency mapping rather than only on additional parameters or an arbitrary fixed per-token feature, we evaluate three control conditions at the 150M scale:

- **Wider Baseline** (d_{model}=792): A baseline with no frequency prior but parameter-matched to FAA.
- **Shuffled Prior**: We randomly permute the mapping between token IDs and their log-probabilities, destroying the token-frequency correspondence while preserving the distributional shape of $I(t)$.
- **Random Prior**: We bootstrap-resample $I(t)$ from the empirical unigram distribution, destroying the token-frequency correspondence while preserving the marginal shape only in expectation.

Table 3 reports the wider-baseline and fixed-prior controls; Table 2 supplies the IG-only reference.

The fixed shuffled and random priors should be compared directly to IG-only, because all three include the gate capacity but lack an aligned corpus-frequency mapping. Under this comparison, Shuffled (11.13 PPL) closely matches IG-only (11.14), and Random (11.19) is slightly worse; fixed unaligned prior-shaped features therefore add approximately no benefit beyond the gate. The corpus-derived prior still reaches 10.79 PPL, giving an additional 0.34–0.40 PPL reduction beyond these unaligned controls and supporting the aligned token-frequency mapping as the source of the frequency-specific gain.

Prior assignment and IWVS direction. We test alternative token assignments while preserving the corpus self-information values. Let $I_{(1)} \leq \cdots \leq I_{(M)}$ be the sorted values for the M non-special tokens. Sorting these tokens by a feature $f(t)$, with ties broken by vocabulary ID, gives the reassignment $I_f(t_{(k)}) = I_{(k)}$. The vocabulary-ID variant uses $f(t) = \mathrm{ID}(t)$, not corpus-frequency rank. The character-entropy variant uses $f(t) = H_{\mathrm{char}}(t) = -\sum_c p_c(t) \log p_c(t)$, where $p_c(t)$ is the character fraction in the stored tokenizer string; entropy is zero for empty or single-character strings. Both leave special-token entries unchanged. They yield similar PPL but lower ARC-E and ARC-C accuracy than the corpus assignment (Table 20), supporting the token–prior correspondence rather than a change in the marginal value distribution. Neither variant uses document frequency or an IDF estimator.

Allowing bidirectional IWVS replaces the amplify-only response with $\gamma = \gamma_{\min} + (1 + s_{\max} - \gamma_{\min})\sigma(\cdot)$ using $\gamma_{\min}$=0.1 and $s_{\max}$=9; non-special token scales lie strictly between $\gamma_{\min}$ and $1 + s_{\max}$. This permits downscaling below identity as well as amplification, but never sign-flips the value contribution. This variant does not improve PPL over amplification-only and yields small gains on PIQA/ARC-C only, consistent with our design choice to keep routing decisions in FAB/softmax and to treat IWVS as an amplification-only complement on the value path.

Table 20: **Prior-assignment and IWVS-direction ablations at 150M.** Single-run comparison of the corpus prior, reassigned priors, and bidirectional-IWVS variants.

Configuration	PPL↓	D0↓	ARC-E	ARC-C	HSwag	PIQA	LAMBADA
FAA, unigram (Ours)	**10.79**	**48.02**	**59.47**	25.77	**33.33**	65.67	**34.10**
FAA, character-entropy reassignment	10.81	48.34	58.12	25.00	32.96	65.72	32.31
FAA, vocabulary-ID reassignment	10.81	48.05	55.93	24.91	33.17	64.91	33.20
FAA + bidirectional IWVS ($\gamma \in [0.1, 10]$)	10.82	48.03	57.58	**26.19**	33.13	**66.92**	32.66

Stacking with training-time logit adjustment. To test whether FAA's attention-level prior is complementary to an output-logit-level prior, we retrain FAA *from scratch* with Menon's training-time logit adjustment (Menon et al., 2021) at τ=1.0 and evaluate with the same training-matched $\mathrm{softmax}(z + \tau \log P(t))$ bias used for the logit-adjustment baselines in Table 19. The combination is roughly neutral-to-slightly-worse on language-modeling metrics but worse on downstream reading: PPL is 10.88 (vs. FAA's 10.79, +0.09), D0 PPL 48.77 (vs. 48.02, +0.75), and LAMBADA 32.02% (vs. 34.10%, −2.08, exceeding the multi-seed FAA standard deviation of 0.49 percentage points). The two interventions operate at different stages—FAA shapes *attention* allocation over keys, whereas logit adjustment rescales the *output distribution*—but combining them does not provide consistent gains over FAA alone.

I Learned frequency-path parameter analysis

The representative parameter snapshot is in Figure 2; the complete IWVS response curves are retained below in Figure 11.

I.1 FAIG beta analysis

Figure 6 compares the default elementwise configuration with a minimum-overhead per-head IG + headwise FAIG configuration:

- **Default elementwise**: elementwise IG with $\mathrm{MLP}_{\mathrm{FAIG}} : \mathbb{R}^1 \to \mathbb{R}^{H d_v}$ and one scalar β_ℓ per layer (12 β parameters for 150M)
- **Minimum-overhead headwise**: per-head IG with $\mathrm{MLP}_{\mathrm{FAIG}} : \mathbb{R}^1 \to \mathbb{R}^H$ broadcast over d_v and one $\beta_{\ell,h}$ per head per layer (144 β parameters for 150M)

The learned β_ℓ (or $\beta_{\ell,h}$) rescales the additional FAIG bias term in the gate logits, controlling its magnitude relative to the content projection XW_{IG}.

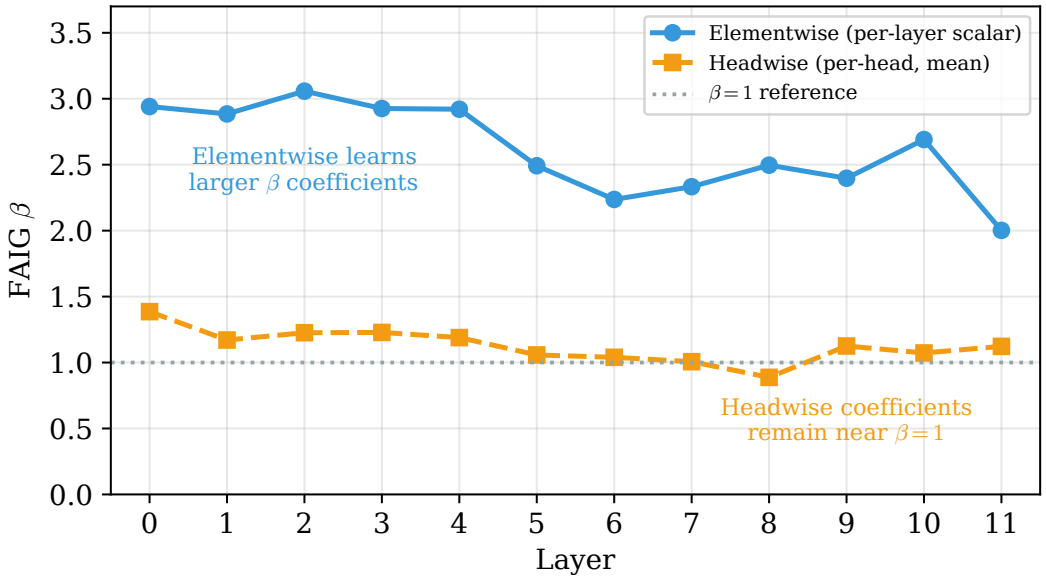


Figure 6: Elementwise and headwise FAIG β values. The default elementwise configuration gives lower PPL than the minimum-overhead per-head IG + headwise FAIG variant (10.79 vs. 11.02).

The default elementwise configuration learns $\beta_\ell \approx 2.0$–3.1 across layers, while the minimum-overhead per-head IG + headwise FAIG configuration learns $\beta_{\ell,h}$ values near $\sim$1.1 on average. The default elementwise configuration also achieves lower PPL than the headwise configuration in Table 2. Because the headwise run also changes IG granularity, we interpret it as a low-overhead configuration comparison rather than a pure FAIG-granularity ablation.

Scalar β values alone do not determine the functional effect of FAIG because the MLP response can vary in sign and magnitude across the frequency spectrum. Figure 7 compares mean post-sigmoid gate values, $\Delta g = \mathbb{E}[g_{\text{FAIG}}] - \mathbb{E}[g_{\text{IG}}]$, between separately trained Full FAA and FAB+IWVS+IG models, aggregated by frequency decile. The difference is mixed-sign and layer-dependent: for example, Layer 7 shows a broadly positive difference across deciles ($+0.027$ to $+0.045$), Layer 9 shows a broadly negative difference (-0.054 to -0.043), and Layer 11 shows a smaller but consistently positive difference ($+0.014$ to $+0.017$). This pattern is consistent with layer-dependent gate calibration rather than a uniform increase for rare tokens.

Headwise gate diagnostic. Although this low-overhead per-head IG + headwise FAIG configuration underperforms the default elementwise configuration, its learned FAIG coefficients vary across heads (Figure 8):

- **L0**: Coefficients approximately 1.33–1.46.
- **L1–L3**: Coefficients mostly above 1, with variation across heads.
- **L4–L8**: Coefficients vary substantially, including values below 0.5 and above 1.5.
- **L9–L11**: Coefficients generally around 1.0–1.2.

The learned β coefficients vary across heads, but this low-overhead configuration does not improve PPL over the default elementwise configuration.

Raw values. Table 21 presents learned FAIG β_ℓ for the elementwise configuration.

Table 22 presents the 12×12 matrix for the headwise configuration.

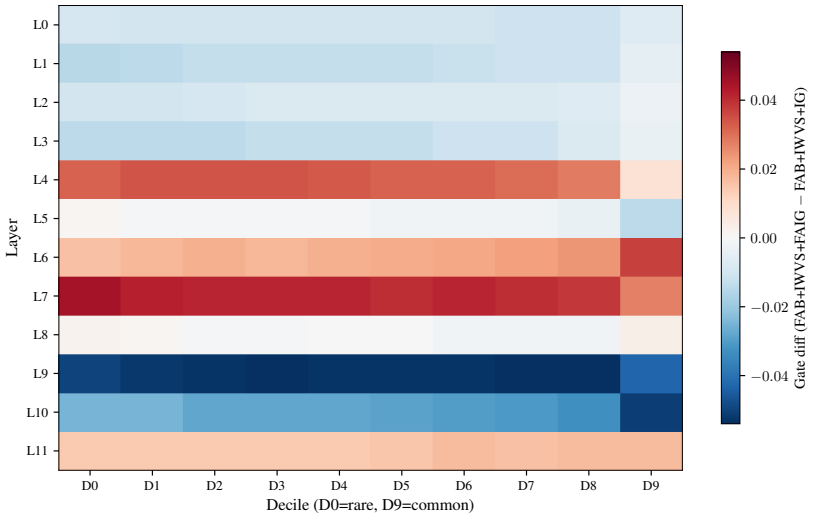


Figure 7: Difference in mean gate activations between separately trained Full FAA and FAB+IWVS+IG models, by decile and layer.

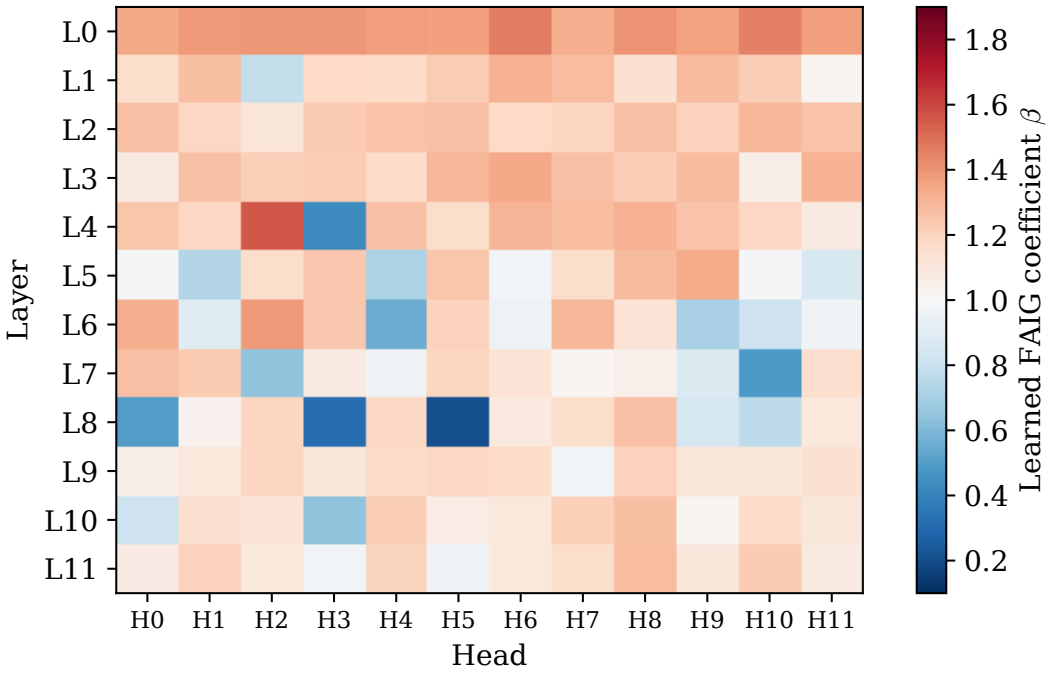


Figure 8: FAIG $\beta_{\ell,h}$ values for the headwise variant. Red denotes larger coefficients; blue denotes smaller coefficients.

Table 21: **FAIG β_ℓ values** (elementwise, one per layer). Learned coefficients multiplying the FAIG MLP output.

Layer	β	**Layer**	β
L0	2.941	L6	2.236
L1	2.885	L7	2.333
L2	3.058	L8	2.496
L3	2.926	L9	2.397
L4	2.921	L10	2.691
L5	2.492	L11	2.002
Mean		2.615	

Table 22: Headwise FAIG $\beta_{\ell,h}$ (150M). Bold values are below the initialization reference of 1.

	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11
L0	1.34	1.38	1.39	1.39	1.37	1.37	1.46	1.33	1.40	1.36	1.45	1.37
L1	1.16	1.27	**0.78**	1.18	1.16	1.23	1.32	1.27	1.14	1.28	1.22	1.03
L2	1.26	1.19	1.11	1.24	1.26	1.27	1.18	1.19	1.26	1.21	1.30	1.26
L3	1.08	1.26	1.22	1.23	1.16	1.30	1.34	1.26	1.22	1.28	1.06	1.32
L4	1.25	1.19	1.56	**0.43**	1.26	1.16	1.30	1.28	1.32	1.25	1.19	1.08
L5	**0.98**	**0.73**	1.16	1.24	**0.72**	1.25	**0.98**	1.16	1.28	1.33	**0.99**	**0.86**
L6	1.33	**0.89**	1.38	1.25	**0.55**	1.21	**0.95**	1.30	1.12	**0.71**	**0.82**	**0.97**
L7	1.26	1.23	**0.64**	1.08	**0.97**	1.19	1.12	1.02	1.05	**0.87**	**0.48**	1.15
L8	**0.50**	1.04	1.20	**0.31**	1.19	**0.21**	1.09	1.16	1.26	**0.85**	**0.77**	1.09
L9	1.06	1.10	1.19	1.11	1.17	1.19	1.16	**0.98**	1.21	1.11	1.10	1.14
L10	**0.82**	1.15	1.11	**0.64**	1.22	1.07	1.09	1.21	1.27	1.01	1.17	1.10
L11	1.07	1.21	1.10	**0.97**	1.20	**0.97**	1.09	1.16	1.28	1.10	1.23	1.08

I.2 FAB ALPHA ANALYSIS

Figure 9 summarizes learned FAB $\alpha_{\ell,h}$ values. Under the multiplicative view (Eq. (4)), each $\alpha_{\ell,h}$ acts as a per-head inverse-frequency exponent—the values below thus characterize the reweighting shape each head learns. The distribution has mean 0.0247, standard deviation 0.0399, and 88.2% of layer–head pairs learn positive values (rare-token preference).

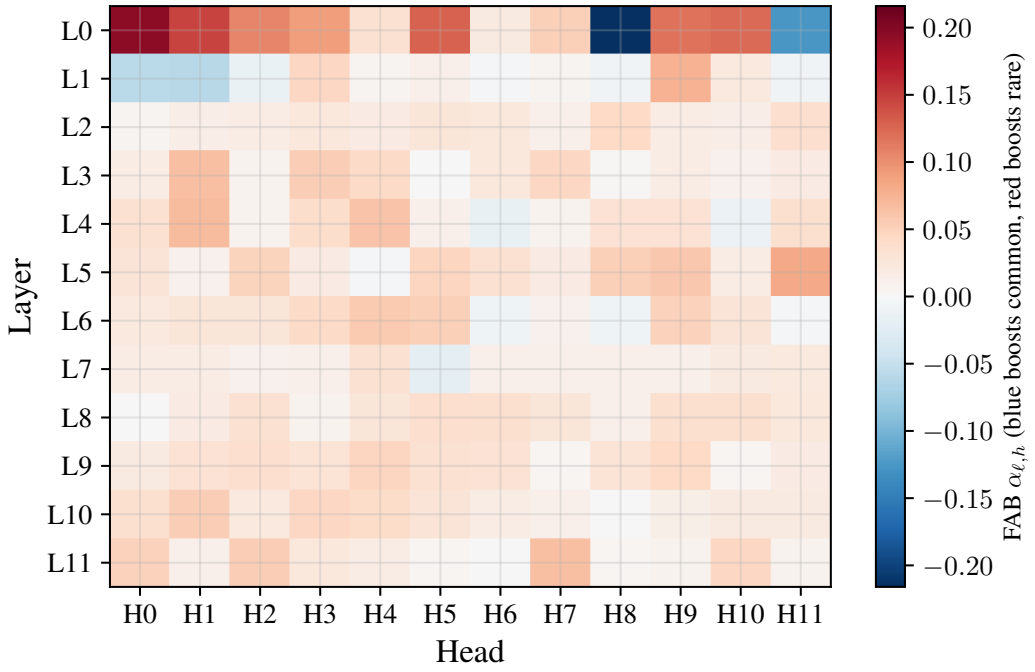


Figure 9: Learned FAB $\alpha_{\ell,h}$ values (150M). 88.2% of layer–head pairs learn positive values.

Layerwise patterns. Distinct patterns emerge:

- **L0**: High variance (range: -0.22 to $+0.19$), with some heads favoring common keys and others favoring rare keys.
- **L1–L9**: Predominantly small positive values ($\sim$0.01–0.04), indicating a consistent, moderate preference for rare tokens.
- **L10–L11**: Consistent positive values with low variance, indicating a stable rare-key preference in later layers.

In this view, $\alpha_{\ell,h} > 0$ upweights rare keys, $\alpha_{\ell,h} < 0$ upweights common keys, and $\alpha_{\ell,h} \approx 0$ approaches content-only attention for that head.

This pattern is consistent with prior analyses of multi-head self-attention showing layerwise specialization (Rogers et al., 2020) and headwise specialization (Voita et al., 2019) in Transformers.

Raw values. Table 23 presents the complete 12×12 matrix of FAB $\alpha_{\ell,h}$ values. Positive indicates rare-token preference; negative indicates common-token preference.

Table 23: **FAB $\alpha_{\ell,h}$ values** across 12 layers × 12 heads. Positive values favor rare keys; bold marks values above 0.05.

	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	Mean
L0	**+0.193**	**+0.147**	**+0.106**	**+0.091**	+0.034	**+0.127**	+0.019	**+0.051**	−0.216	**+0.117**	**+0.122**	−0.126	+0.055
L1	−0.059	−0.060	−0.014	+0.045	+0.006	+0.011	−0.004	+0.006	−0.007	**+0.076**	+0.021	−0.007	+0.001
L2	+0.006	+0.014	+0.016	+0.023	+0.018	+0.027	+0.023	+0.010	+0.043	+0.016	+0.014	+0.036	+0.020
L3	+0.016	**+0.066**	+0.008	**+0.053**	+0.042	+0.001	+0.022	+0.045	+0.002	+0.017	+0.009	+0.018	+0.025
L4	+0.033	**+0.068**	+0.007	+0.038	**+0.062**	+0.011	−0.016	+0.008	+0.032	+0.032	−0.013	+0.035	+0.025
L5	+0.028	+0.010	+0.049	+0.018	−0.003	+0.047	+0.033	+0.017	**+0.052**	**+0.058**	+0.017	**+0.082**	+0.034
L6	+0.021	+0.026	+0.026	+0.041	**+0.057**	**+0.051**	−0.009	+0.010	−0.010	+0.050	+0.028	−0.002	+0.024
L7	+0.017	+0.017	+0.010	+0.010	+0.032	−0.022	+0.010	+0.011	+0.011	+0.012	+0.019	+0.021	+0.012
L8	+0.000	+0.018	+0.033	+0.008	+0.026	+0.036	+0.035	+0.027	+0.011	+0.035	+0.035	+0.022	+0.024
L9	+0.019	+0.032	+0.036	+0.029	+0.046	+0.032	+0.032	+0.004	+0.029	+0.043	+0.004	+0.017	+0.027
L10	+0.034	**+0.052**	+0.021	+0.045	+0.040	+0.028	+0.016	+0.011	−0.001	+0.012	+0.020	+0.019	+0.025
L11	+0.050	+0.011	**+0.054**	+0.023	+0.015	+0.004	−0.001	**+0.064**	+0.005	+0.007	+0.045	+0.008	+0.024

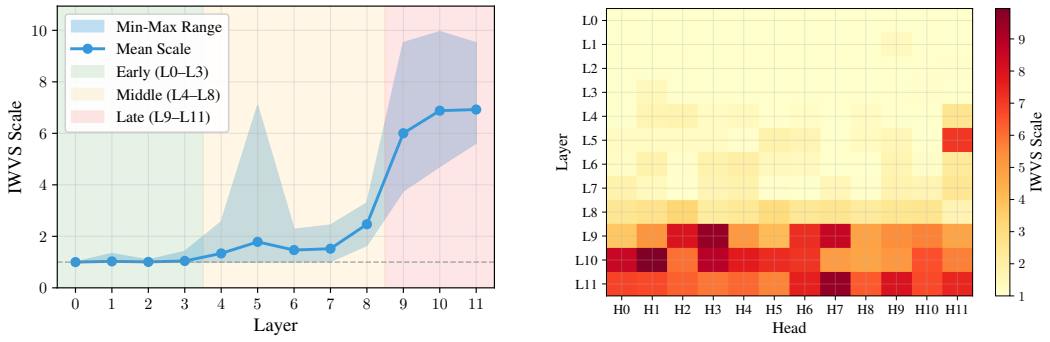
I.3 IWVS SCALE ANALYSIS

Depth-dependent mean scale on a representative evaluation batch. Because IWVS defines an amplification-only token-dependent scale $\gamma_{\ell,h}(\bar{I}(t))$ upper-bounded by $1+s_{\max}$, we summarize a single per-head value by averaging the scale actually applied to tokens in a representative held-out FineWeb-Edu evaluation batch: $\mathbb{E}_{b,t}[\gamma_{\ell,h}(\bar{I}(t_{b,t}))]$. This is the representative evaluation-batch mean scale, not the response at $\bar{I}{=}1$ or the maximum of $\gamma_{\ell,h}(\cdot)$ over all inputs. We observe a strong increase in this mean scale with depth:

- **L0–L3**: mean scale ≈ 1.0 (near-unity response)
- **L4–L8**: mean scale ≈ 1.3–2.5 (moderate)
- **L9–L11**: mean scale ≈ 6–7, reaching ∼10×

Several heads in final layers approach the architectural cap (10×) on this batch, so deep-layer heads can learn strong value amplification when trained end-to-end. Figure 10a summarizes this depth pattern.

Layer vs. head variance. IWVS scale variance across layers (5.01) exceeds variance across heads within layers (0.54) by ∼9×. This suggests that depth explains more variation than head identity in this summary statistic; Figure 10b shows the corresponding per-head pattern.



(a) Per-layer summary (line plot).

(b) Per-head heatmap (12 layers × 12 heads).

Figure 10: Representative evaluation-batch IWVS mean scale $\mathbb{E}_{b,t}[\gamma_{\ell,h}(\bar{I}(t_{b,t}))]$ at 150M. Early layers stay near identity; late layers amplify more.

Raw mean scales. Table 24 presents the complete 12×12 matrix of representative evaluation-batch mean scales.

Table 24: **IWVS representative evaluation-batch mean scale.** Values average $\gamma_{\ell,h}(\bar{I}(t))$ over observed tokens; bold marks scales ≥ 2.0.

	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	Mean
L0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
L1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.33	1.00	1.00	1.03
L2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.08	1.01
L3	1.00	1.41	1.00	1.02	1.00	1.00	1.00	1.00	1.00	1.00	1.09	1.00	1.04
L4	1.00	1.68	1.72	1.27	1.37	1.00	1.20	1.00	1.24	1.00	1.00	**2.56**	1.34
L5	1.21	1.28	1.27	1.24	1.07	1.79	1.64	1.00	1.31	1.54	1.00	**7.07**	1.78
L6	1.14	1.79	1.08	1.83	1.95	1.39	1.34	1.00	1.01	1.70	1.11	**2.27**	1.47
L7	1.82	1.38	1.03	1.66	1.80	1.07	1.02	1.55	1.08	1.76	1.64	**2.44**	1.52
L8	**2.44**	**2.65**	**3.30**	**2.14**	**2.34**	**3.01**	**2.39**	**2.52**	**2.22**	**2.46**	**2.53**	1.65	2.47
L9	**3.76**	**5.20**	**7.93**	**9.52**	**5.11**	**4.10**	**7.27**	**8.54**	**4.79**	**5.42**	**5.66**	**4.75**	6.00
L10	**8.49**	**9.95**	**5.98**	**8.87**	**7.69**	**7.36**	**7.12**	**5.01**	**4.71**	**5.12**	**6.57**	**5.69**	6.88
L11	**6.83**	**6.71**	**6.25**	**5.89**	**6.14**	**5.62**	**7.60**	**9.52**	**6.35**	**8.02**	**6.70**	**7.46**	6.92

Pre-sigmoid logit analysis. Table 25 reports the pre-sigmoid logits $z_{\ell,h}(\bar{I})$ at common, mid, and rare probe inputs, with panels (a) and (b) covering L0–L5 and L6–L11, respectively; applying $\gamma_{\ell,h} = 1 + s_{\max} \cdot \sigma(z_{\ell,h})$ gives the response curves in Figure 11. These probe-input responses are separate from the representative evaluation-batch mean scales in Table 24, which average $\gamma_{\ell,h}(\bar{I}(t))$ over observed tokens. L0 logits are strongly negative across the probed inputs, yielding scales effectively equal to one. In this run, the first layer therefore does not use frequency-dependent value scaling; the learned IWVS response emerges mainly in deeper layers.

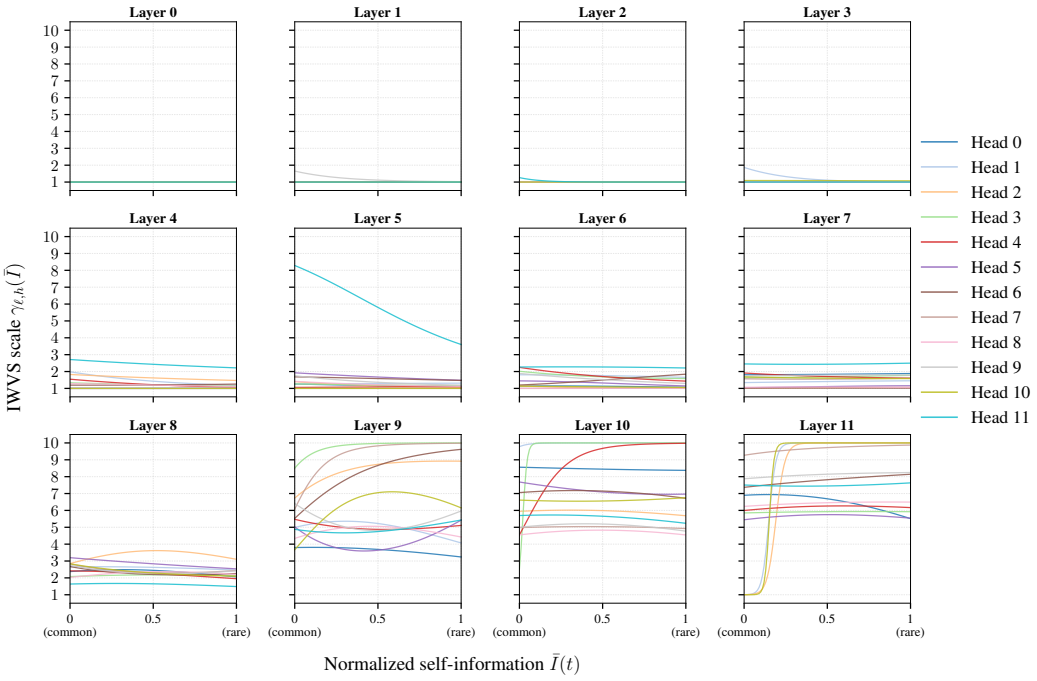


Figure 11: IWVS response $\gamma_{\ell,h}(\bar{I})$ for all layers; later layers learn more varied scaling.

Table 25: IWVS pre-sigmoid logits $z_{\ell,h}(\bar{I})$ at $\bar{I} \in \{0, 0.5, 1\}$ (common to rare).

(a) Layers L0–L5.

Layer	$\bar{I}$	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11
L0	0.0	−17.8	−16.7	−16.9	−17.5	−17.3	−16.9	−16.0	−16.8	−16.1	−16.2	−16.0	−16.2
	0.5	−25.8	−24.2	−24.5	−25.3	−25.0	−24.5	−23.2	−24.3	−23.4	−23.5	−23.2	−23.5
	1.0	−33.9	−31.9	−32.2	−33.3	−32.9	−32.2	−30.5	−32.0	−30.9	−30.9	−30.5	−30.9
L1	0.0	−12.6	−14.6	−14.7	−14.4	−14.8	−14.1	−14.9	−14.8	−14.7	−2.5	−13.9	−14.7
	0.5	−11.9	−15.2	−15.2	−16.2	−15.2	−15.6	−15.7	−15.1	−15.6	−4.3	−16.0	−15.1
	1.0	−11.9	−16.1	−16.0	−18.3	−15.9	−17.3	−16.9	−15.7	−16.8	−6.0	−18.3	−15.8
L2	0.0	−11.3	−12.4	−12.6	−11.9	−10.0	−11.0	−12.6	−11.2	−11.0	−12.5	−12.4	−3.5
	0.5	−23.1	−25.4	−24.2	−24.4	−19.9	−22.5	−24.4	−23.5	−22.3	−24.6	−25.4	−8.2
	1.0	−36.8	−40.4	−37.4	−38.8	−31.3	−35.9	−37.8	−37.6	−35.3	−38.4	−40.4	−13.7
L3	0.0	−13.1	−2.2	−13.7	−5.4	−13.7	−13.4	−14.0	−14.2	−13.8	−11.0	−4.6	−14.5
	0.5	−17.6	−4.4	−17.6	−6.7	−16.6	−18.2	−17.6	−17.6	−17.5	−13.9	−4.7	−18.6
	1.0	−22.6	−6.7	−21.9	−8.2	−19.8	−23.5	−21.6	−21.3	−21.6	−17.1	−4.8	−23.1
L4	0.0	−12.9	−2.1	−2.3	−3.3	−2.7	−10.5	−3.8	−14.4	−3.4	−14.1	−14.5	−1.4
	0.5	−16.2	−3.0	−2.6	−3.7	−3.7	−9.4	−3.7	−15.0	−3.8	−10.2	−15.9	−1.7
	1.0	−19.7	−3.9	−2.9	−4.2	−4.9	−9.1	−3.5	−15.9	−4.4	−6.6	−17.4	−1.9
L5	0.0	−3.5	−3.5	−3.1	−3.5	−5.0	−2.2	−2.5	−13.4	−3.0	−2.4	−11.8	1.4
	0.5	−4.0	−3.4	−4.0	−3.7	−4.6	−2.5	−2.7	−12.9	−3.7	−3.2	−11.9	0.1
	1.0	−4.4	−3.3	−4.9	−4.0	−4.5	−2.9	−2.9	−13.0	−4.3	−3.8	−12.4	−0.9

(b) Layers L6–L11.

Layer	$\bar{I}$	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11
L6	0.0	−4.1	−2.3	−4.0	−2.1	−1.8	−3.0	−3.8	−12.2	−7.9	−2.2	−4.2	−1.8
	0.5	−4.2	−2.4	−5.6	−2.5	−2.5	−3.3	−2.8	−8.5	−6.9	−2.8	−4.6	−1.8
	1.0	−4.8	−2.5	−7.1	−2.7	−3.0	−4.1	−2.3	−5.5	−6.0	−3.4	−5.0	−1.9
L7	0.0	−2.3	−3.2	−6.2	−2.6	−2.2	−5.4	−5.9	−2.7	−4.9	−2.4	−2.5	−1.6
	0.5	−2.3	−3.0	−5.2	−2.5	−2.5	−4.4	−6.1	−2.7	−4.7	−2.3	−2.6	−1.7
	1.0	−2.2	−2.9	−5.0	−2.4	−2.6	−4.0	−6.3	−2.6	−5.0	−2.3	−2.7	−1.6
L8	0.0	−1.7	−1.5	−1.4	−2.0	−1.7	−1.1	−1.5	−1.4	−2.1	−1.4	−1.4	−2.6
	0.5	−1.7	−1.5	−0.9	−1.9	−1.8	−1.4	−1.9	−1.8	−1.7	−1.8	−1.8	−2.6
	1.0	−2.0	−1.6	−1.2	−1.9	−2.1	−1.6	−1.8	−1.7	−1.9	−1.7	−2.0	−2.9
L9	0.0	−0.8	−0.2	0.6	1.6	−0.0	−0.2	0.0	0.3	−0.5	0.4	−0.9	−0.3
	0.5	−0.9	−0.1	1.8	5.7	−0.3	−0.9	1.7	3.9	−0.2	−0.3	0.7	−0.3
	1.0	−1.1	−0.7	2.0	8.8	−0.2	−0.0	3.1	6.3	−0.5	0.2	0.3	−0.0
L10	0.0	1.7	3.7	0.2	−1.6	−0.4	1.1	0.7	−0.2	−0.4	−0.2	0.5	0.1
	0.5	1.6	12.3	0.2	32.7	3.3	0.7	0.8	−0.2	−0.3	−0.1	0.5	0.1
	1.0	1.5	19.0	0.1	57.9	6.1	0.7	0.6	−0.3	−0.4	−0.3	0.6	−0.1
L11	0.0	0.6	−7.8	−7.9	0.2	0.2	−0.0	0.9	2.4	0.3	1.2	−12.2	1.0
	0.5	0.5	13.8	9.8	0.2	0.3	0.1	1.1	3.4	0.4	1.3	21.1	0.9
	1.0	0.0	24.5	19.5	0.2	0.3	0.0	1.3	4.2	0.4	1.4	37.1	1.0

Scale parameter ablation. Table 26 compares $s_{\max} \in \{4, 9\}$. We select $s_{\max} = 9$ based on stronger language-modeling metrics (PPL and D0 PPL) and overall downstream performance. These settings bracket moderate ($s_{\max}$=4, initial mean scale $1 + s_{\max}/2$=3.0×) and aggressive ($s_{\max}$=9, initial mean scale 5.5×) amplification under our 25k-vocabulary unigram prior.

Table 26: IWVS $s_{\max}$ ablation (150M). $s_{\max} = 9$ gives lower PPL/D0 and higher scores on four of five downstream tasks.

Metric	$s_{\max} = 4$	$s_{\max} = 9$
PPL ↓	10.90	**10.79**
D0 PPL ↓	49.23	**48.02**
ARC-E ↑	57.70%	**59.47%**
ARC-C ↑	24.06%	**25.77%**
HSwag ↑	32.97%	**33.33%**
PIQA ↑	**66.81%**	65.67%
LAMBADA ↑	31.21%	**34.10%**

$s_{\max} = 9$ provides 1.0% lower PPL and lower D0 PPL (49.23→48.02), with a particularly strong improvement of 2.89 percentage points on LAMBADA where rare proper nouns are critical.

Stability under large γ. Because IWVS can amplify per-head value norms by up to $1+s_{\max}$=10×, we check whether this causes training divergence. Across all three 150M seeds, FAA training loss decreases without divergence spikes (Figure 3). This stability check is nontrivial because from-scratch IWVS is not identity-initialized for non-special tokens (Appendix D); with $s_{\max}$=9, the initial scale is centered near $1 + s_{\max}/2 = 5.5$. After training, early layers settle near unity (L0–L3), while heads that approach the 10× cap are concentrated in late layers (L9–L11; Figure 10a).

J Extended related work

Positional priors in attention. Our LLaMA-style backbone uses RoPE (Su et al., 2024), which rotates queries and keys as a function of position to encode *where* tokens appear. Other positional attention priors, including ALiBi (Press et al., 2022) and the relative position bias used in T5 (Raffel et al., 2020), similarly encode position or distance structure rather than vocabulary identity. FAA is orthogonal: it supplies tokenizer-indexed corpus statistics to attention pathways, encoding *which* vocabulary item is present rather than its position.

Frequency-aware methods in natural language processing. Gong et al. (2018) propose FRAGE, using adversarial training to reduce frequency-related separability in embeddings (i.e., blurring the boundary between high- and low-frequency embedding clusters), making them "frequency-agnostic." Other prior work instead uses frequency statistics at the output-logit level through logit adjustment (Menon et al., 2021) or analyzes how Transformer prediction heads encode word-frequency effects (Kobayashi et al., 2023). At the tokenizer level, long-token-first byte-pair encoding (LBPE) (Lian et al., 2025) changes the encoding procedure itself to prioritize longer tokens, thereby reshaping the frequency spectrum before language-model training begins. As discussed in Section 2, FAA instead injects frequency information inside attention while keeping the tokenizer fixed. Distinct intervention points do not imply constructive composition: stacking FAA with training-time logit adjustment is neutral-to-slightly-worse on PPL/D0 and worse on LAMBADA rather than additive (Appendix H.2, stacking analysis), suggesting FAA already captures the frequency-aware signal accessible to a unigram prior.

Chang & Bergen (2022) show that language models depend more strongly on word frequency than children do, and that models predict from unigram frequencies early in training before gradually transitioning to bigram probabilities. A complementary line argues that token-frequency structure itself is a useful signal for pre-training: Chung & Kim (2025) show that enlarging the vocabulary amplifies token-frequency imbalance and reduces pre-training loss primarily by reducing uncertainty on frequent words, and He et al. (2025) report that models peak when token distributions adhere closely to Zipf's law.

Value path and attention expressivity. SIF-style sentence embeddings motivate aggregation-time inverse-frequency weighting by downweighting frequent words before averaging (Arora et al., 2017); IWVS transfers only this aggregation-side intuition to the Transformer value path. Prior analyses show that attention outputs depend on both routing weights and transformed value vectors. Kobayashi et al. (2020) show that attention weights alone are insufficient to explain token contribution; in their encoder-only BERT analysis, transformed-value norms are suppressed for high-frequency tokens, with Spearman $\rho \approx 0.69$ between frequency rank and $\|f(x)\|$ after excluding `[CLS]`, `[SEP]`, periods, and commas. Gupta & Berant (2021) likewise show that sparse-attention approximations accounting for value vectors recover attention outputs more accurately than weight-only approximations. Attention weights need not be uniquely determined by the attention output (Brunner et al., 2020), so our interpretation focuses on controlled architectural interventions rather than weight-only attribution. Complementarily, Dong et al. (2021) prove that pure self-attention has a strong bias toward token uniformity and rank collapse, while skip connections and MLPs counteract this degeneration. We do not claim that these analyses derive IWVS directly; rather, they motivate intervening on the value pathway rather than only adding another routing bias.

Outlier features and KV cache. Dettmers et al. (2022) identify large-magnitude outlier feature dimensions that emerge systematically in large language models, appear in the inputs to Transformer projection layers, and dominate attention and predictive performance. Sun et al. (2024) document "massive activations" that can dominate downstream computations. These works show that Transformer computations can concentrate on a small subset of features or positions. FAA targets a different concentration point inside attention: it reweights keys with FAB and rescales value contributions with IWVS using static corpus statistics. SnapKV (Li et al., 2024) selects clustered important KV positions per head, PyramidKV (Cai et al., 2025) additionally allocates per-layer KV budgets that shrink with depth, and FASA (Wang et al., 2026) uses RoPE frequency-chunk sparsity to predict salient tokens for focused long-context attention; these methods compress or sparsify past-position storage or computation at runtime, whereas FAA changes attention computation with a static corpus-frequency prior and without runtime KV selection.

K ATTENTION VISUALIZATION

Figure 12 illustrates head-averaged attention at Layers 1 and 6 for a single probe sentence. We report the tokenization, quantitative statistics, and all-layer maps here; pooled attention across three seeds is analyzed in Appendix S.1.

Sentence analysis. The probe sentence tokenizes to 11 tokens (`[sp]` denotes the tokenizer's leading-whitespace marker):

```
The, [sp]mitochondria, [sp]is, [sp]the, [sp]power,
house, [sp]o, f, [sp]the, [sp]cell, .
```

The low-frequency token `[sp]mitochondria` falls in decile D1 under our evaluator-compatible tokenizer deciles; the remaining scored subword tokens are in D8–D9. Table 27 shows the complete decile assignment for each token.

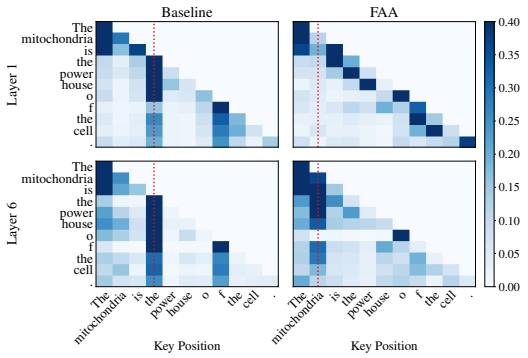


Figure 12: **Head-averaged attention maps (150M).** Rows are queries and columns are keys for "The mitochondria is the powerhouse of the cell." Layer 1 provides a low-layer reference, and Layer 6 shows FAA's shift toward the rare key *mitochondria*; all-layer maps appear in Figures 13 and 14.

Table 27: Probe-sentence token deciles for the attention-visualization example. Deciles use the same subword-only convention as the PPL evaluator; under the FineWeb-Edu prior, `[sp]mitochondria` (pos. 1) falls in D1.

Pos	Token	Decile	Category	$\log P$
0	`The`	D9	Common	−6.33
1	`[sp]mitochondria`	D1	Rare	−12.41
2	`[sp]is`	D9	Common	−4.78
3	`[sp]the`	D9	Common	−3.35
4	`[sp]power`	D9	Common	−8.13
5	`house`	D8	Common	−10.23
6	`[sp]o`	D9	Common	−3.81
7	`f`	–	Byte	−3.81
8	`[sp]the`	D9	Common	−3.35
9	`[sp]cell`	D9	Common	−8.92
10	`.`	–	Byte	−3.40

Quantitative statistics. Table 28 reports the Layer 6 probe aggregate.

Table 28: L6 attention statistics for the sentence in Figure 12, averaged over 12 heads and causally eligible query positions. "Mitochondria" is key position 1.

Metric	Baseline	FAA
Attention to "mitochondria" (pos. 1)	12.8%	28.7%
Rare-token boost	2.2×	

In this probe example, Table 28 shows that FAA increases attention to "mitochondria" (key pos. 1) from 12.8% to 28.7% (2.2× boost).

L6 attention matrices. Figure 12 shows the head-averaged Layer 6 matrices for both Baseline and FAA. Each mean below averages one key column over its causally eligible queries.

In this probe, the Baseline's largest key-column mean is at pos. 3 ("the"; mean 0.452), followed by pos. 0 (mean 0.328). The high attention to pos. 3 may reflect reliance on a frequent structural token, but this single example is not a corpus-level sink or frequency-bias estimate.

FAA allocates more attention mass to pos. 1 ("mitochondria"; mean 0.287) while retaining attention to pos. 0 (mean 0.303). The shift from pos. 3 to pos. 1 is a qualitative example of FAA's key-side routing effect; attention weights alone do not isolate IWVS value-path contributions, which are analyzed separately in Appendix I.3. Attention to "the" (pos. 3) drops from 0.452 to 0.084, a **5.4× reduction**. All-layer maps for L0–L5 and L6–L11 are shown in Figures 13 and 14.

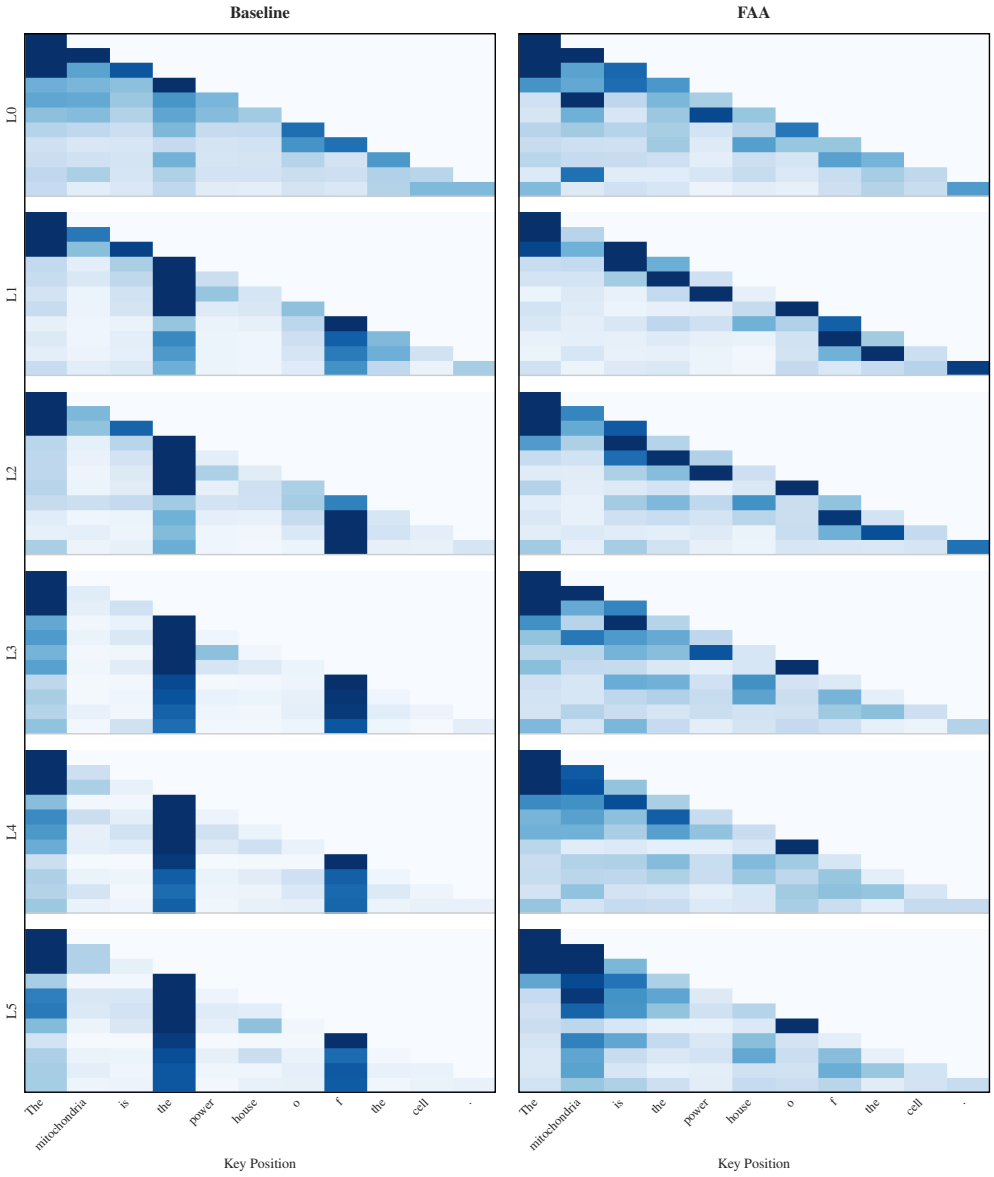


Figure 13: All-layer head-averaged attention maps for L0–L5 (rows: queries; columns: keys). Baseline (left) vs. FAA (right).

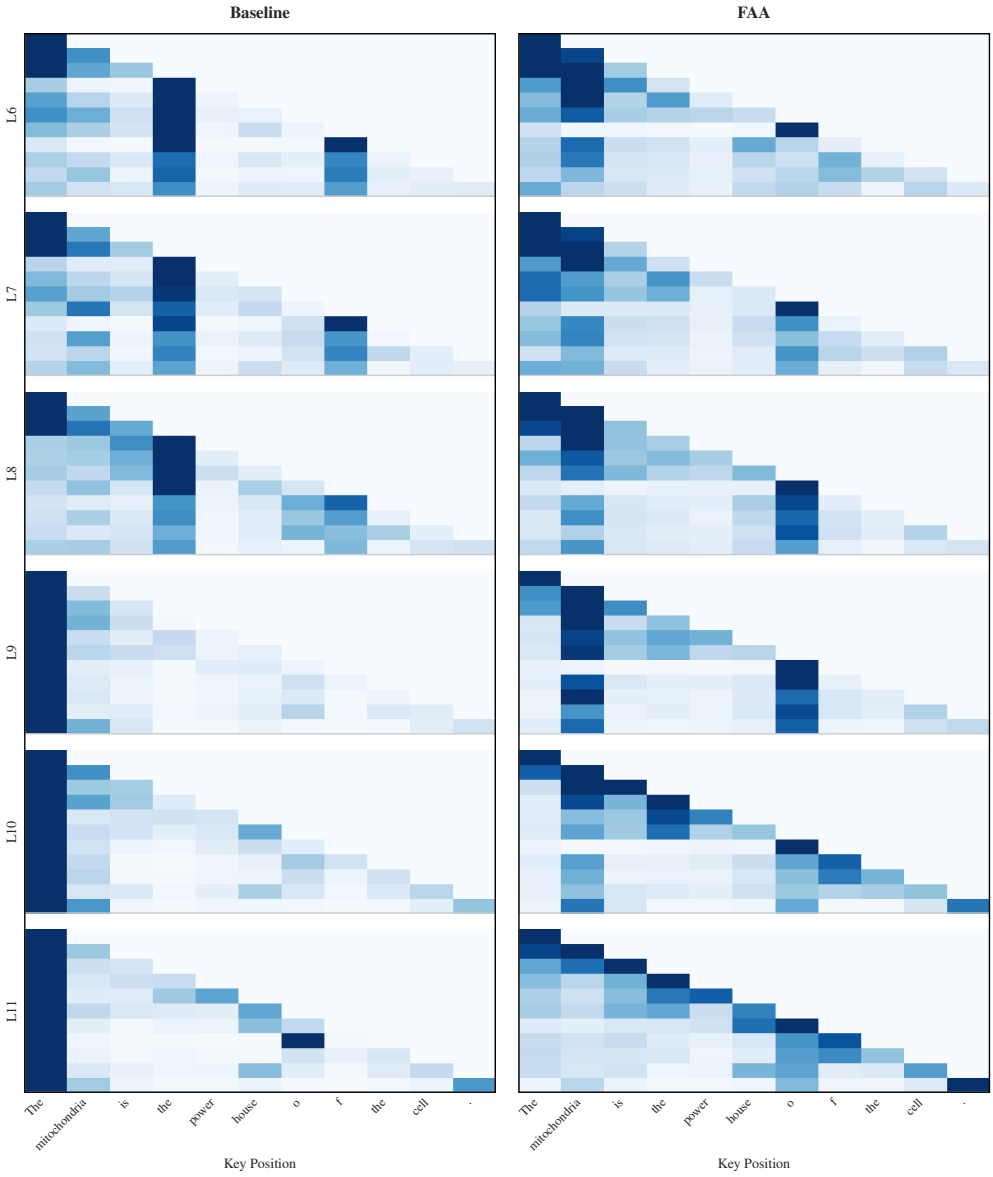


Figure 14: All-layer head-averaged attention maps for L6–L11 (rows: queries; columns: keys). Baseline (left) vs. FAA (right).

L MULTI-SEED VALIDATION

To quantify seed variability at the main 150M scale, we trained both models with three random seeds. Table 29 presents comprehensive results. Across-seed standard deviations describe the observed runs using the population convention (n denominator). Quality means and standard deviations are rounded to two decimal places after aggregation; the shared-token and GPT-NeoX comparisons use the same convention.

Table 29: 150M multi-seed validation. Mean ± standard deviation across 3 seeds; FAA lowers PPL/D0 in every seed and improves downstream means.

Metric	Baseline		FAA		Δ	Rel. Δ
	Mean ± standard deviation	[min, max]	Mean ± standard deviation	[min, max]		
PPL ↓	11.36 ± 0.02	[11.34, 11.39]	**10.80 ± 0.01**	[10.79, 10.82]	−0.56	−4.9%
D0 PPL ↓	53.60 ± 0.14	[53.46, 53.79]	**48.85 ± 0.59**	[48.02, 49.30]	−4.75	−8.9%
ARC-E ↑	56.96 ± 1.20	[55.26, 57.83]	**58.85 ± 0.54**	[58.16, 59.47]	+1.89	+3.3%
ARC-C ↑	23.80 ± 0.66	[23.29, 24.74]	**25.86 ± 0.53**	[25.26, 26.54]	+2.05	+8.6%
HSwag ↑	32.14 ± 0.27	[31.88, 32.52]	**33.27 ± 0.05**	[33.20, 33.33]	+1.13	+3.5%
PIQA ↑	65.56 ± 0.28	[65.29, 65.94]	**66.18 ± 0.47**	[65.67, 66.81]	+0.62	+0.9%
LAMBADA ↑	31.71 ± 0.36	[31.36, 32.21]	**33.50 ± 0.49**	[32.89, 34.10]	+1.79	+5.6%

Key findings.

- **Consistent PPL and D0 PPL reductions**: In this 3-seed sample, PPL and D0 ranges do not overlap; unadjusted parametric paired t-tests give $p = 0.001$ (PPL) and $p = 0.01$ (D0 PPL).
- **Downstream consistency**: FAA improves all three seeds on every benchmark at 150M; unadjusted parametric paired t-tests give $p < 0.05$ for ARC-C ($p = 0.007$) and HellaSwag ($p = 0.03$), with LAMBADA borderline (paired $p = 0.09$, Welch $p = 0.02$). With n=3, we treat the remaining downstream results as descriptive consistency evidence.[2]

L.1 450M AND 1.2B SUMMARY

Table 30: **Scalability results.** 450M reports a two-seed mean; 1.2B is a single run. Per-seed 450M numbers are in Appendix L.2, Table 31.

Scale	Configuration	Params	PPL↓	D0↓	ARC-E	ARC-C	HSwag	PIQA	LAMBADA
450M	Baseline	454.1M	9.29	41.45	62.78	27.61	37.83	68.94	38.89
	FAA (Ours)	480.4M	**8.98**	**38.89**	**65.05**	**30.34**	**38.78**	**70.13**	**42.34**
	Δ	+5.8%	−3.4%	−6.2%	+2.27	+2.73	+0.95	+1.19	+3.45
1.2B	Baseline	1176.6M	8.35	37.53	**69.53**	33.62	41.81	71.98	46.03
	FAA (Ours)	1245.5M	**8.04**	**35.11**	69.07	**35.15**	**44.18**	**72.69**	**50.16**
	Δ	+5.9%	−3.7%	−6.4%	−0.46	+1.53	+2.37	+0.71	+4.13

[2]With n=3, exact two-sided sign-flip tests bottom out at $p = 2/2^3 = 0.25$; all seven metrics reach this floor. Unpaired Welch t-tests give $p < 0.05$ for PPL, D0 PPL, ARC-C, HellaSwag, LAMBADA.

L.2 450M PER-SEED RESULTS

For 450M we additionally trained both Baseline and FAA on a second seed. Table 31 reports per-seed values; the 450M row of Table 30 is the mean of the two. FAA improves PPL, D0 PPL, and every downstream benchmark in both seeds. Descriptively, FAA also has a smaller LAMBADA range across these two seeds (Baseline range ≈2.7 percentage points; FAA range ≈0.4 percentage points), consistent with the hypothesis that LAMBADA stresses rare-token reading.

Table 31: Per-seed 450M validation over two seeds. Table 30 reports the mean.

Metric	Baseline		FAA		Mean Δ	Rel. Δ
	Run A	Run B	Run A	Run B		
PPL ↓	9.33	9.26	**8.98**	**8.98**	−0.31	−3.4%
D0 PPL ↓	41.66	41.25	**38.74**	**39.04**	−2.56	−6.2%
ARC-E ↑	62.67	62.88	**65.11**	**64.98**	+2.27	+3.6%
ARC-C ↑	27.05	28.16	**29.44**	**31.23**	+2.73	+9.9%
HSwag ↑	37.85	37.80	**38.81**	**38.75**	+0.95	+2.5%
PIQA ↑	69.53	68.34	**70.35**	**69.91**	+1.19	+1.7%
LAMBADA ↑	40.23	37.55	**42.52**	**42.15**	+3.45	+8.9%

M DECILE PPL COMPARISON ACROSS SCALES

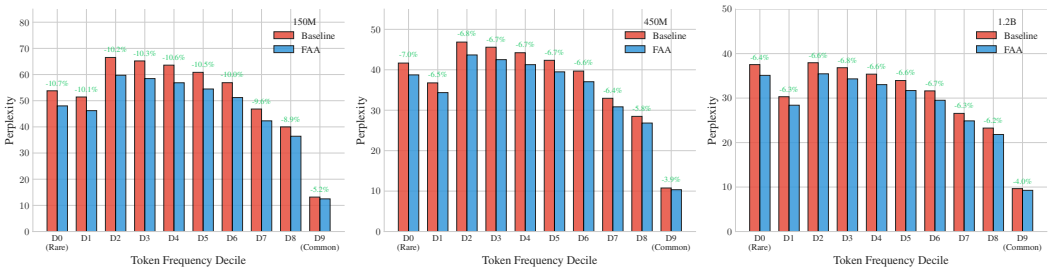


Figure 15: **Decile-wise PPL across 150M, 450M, and 1.2B.** Representative runs; D0 is rarest and D9 most common. FAA lowers every decile, with larger relative gains on low-frequency tokens.

Table 32 reports the occurrence mass of the evaluator-compatible deciles used for D0–D9 PPL: deciles are balanced by vocabulary-item count but extremely imbalanced by occurrence count. Table 33 then presents the complete per-decile PPL comparison across all three scales (150M, 450M, and 1.2B). FAA reduces PPL across all frequency deciles at every scale in the reported runs. The broad reduction is compatible with the hypothesis that improving rare-token representations benefits the full token distribution, aligning with the finding of Yu et al. (2022) that rare-token degeneration can affect all tokens.

Table 32: **Evaluator-compatible decile occurrence mass.** Deciles exclude special and single-byte tokens; columns report non-special corpus-token mass and subword-only mass.

Decile	# Vocab items	% Non-special tokens	% Subword mass
D0	2,476	0.1598	0.2126
D1	2,476	0.8620	1.1469
D2	2,476	1.1586	1.5416
D3	2,477	1.3898	1.8491
D4	2,476	1.6987	2.2602
D5	2,476	2.1577	2.8709
D6	2,477	2.9131	3.8760
D7	2,476	4.3105	5.7353
D8	2,476	7.5966	10.1076
D9	2,477	52.9105	70.3998

Table 33: **Complete decile-wise PPL across scales.** Representative runs; Baseline (BL) and FAA use identical decile boundaries. D0 is rarest and D9 most common.

	150M			450M			1.2B		
Decile	**BL**	**FAA**	**Δ**	**BL**	**FAA**	**Δ**	**BL**	**FAA**	**Δ**
D0 (Rare)	53.79	**48.02**	−10.7%	41.66	**38.74**	−7.0%	37.53	**35.11**	−6.4%
D1	51.42	**46.21**	−10.1%	36.76	**34.36**	−6.5%	30.31	**28.40**	−6.3%
D2	66.53	**59.72**	−10.2%	46.85	**43.67**	−6.8%	37.95	**35.46**	−6.6%
D3	65.20	**58.46**	−10.3%	45.58	**42.51**	−6.7%	36.83	**34.32**	−6.8%
D4	63.59	**56.86**	−10.6%	44.22	**41.27**	−6.7%	35.38	**33.03**	−6.6%
D5	60.84	**54.48**	−10.5%	42.34	**39.49**	−6.7%	33.95	**31.70**	−6.6%
D6	56.89	**51.21**	−10.0%	39.68	**37.05**	−6.6%	31.62	**29.51**	−6.7%
D7	46.82	**42.31**	−9.6%	32.95	**30.84**	−6.4%	26.56	**24.88**	−6.3%
D8	39.99	**36.42**	−8.9%	28.48	**26.83**	−5.8%	23.26	**21.82**	−6.2%
D9 (Common)	13.17	**12.49**	−5.2%	10.77	**10.35**	−3.9%	9.65	**9.26**	−4.0%

N Dataset and tokenizer details

We use the FineWeb-Edu `sample/100BT` release (Penedo et al., 2024), exported locally as 140 Apache Arrow shards. For language-model pre-training we use shards 0–127 as training (128 shards; ≈103.4B tokens under our BBPE tokenizer, following subword byte-pair encoding practice (Sennrich et al., 2016) and BBPE design (Wang et al., 2020)) and reserve shard 139 as evaluation (≈289.0M tokens), leaving shards 128–138 (11 shards) unused. Pre-training uses fixed step budgets rather than validation-based early stopping. Unigram statistics $P(t)$ and quantile floor clamping are computed from the merged tokenizer-training states for all 140 shards: the final BBPE model tokenizes the retained strings, and each resulting token count is weighted by the stored string frequency (113.0B counted token occurrences; see `tokenizer.json` metadata). The tokenizer and its fixed unigram table are shared across the language-model training/evaluation split; language-model pre-training itself uses only the training shards described above, and the reserved final shard is not used for model-parameter updates.

Downstream evaluation. We use `lm-evaluation-harness` (Sutawika et al., 2025) for downstream zero-shot benchmarks on locally mirrored datasets, ensuring reproducibility under offline conditions. The local task configuration files evaluate ARC-E, ARC-C, and LAMBADA on test splits and HellaSwag and PIQA on validation splits. We use the harness's default log-likelihood-of-completions scoring without additional length normalization.

Prior split sensitivity. The language-model training stream excludes the held-out evaluation shard. The reported full-corpus prior nevertheless contains an aggregate unigram-count contribution from shard 139, which accounts for 280,122,684 tokens, or 0.248% of the prior-count mass. To quantify this corpus-statistics effect, we recompute the prior with the same BBPE merge rules and token-frequency code used to build the tokenizer, then compare the resulting token distributions and decile assignments to the reported full prior. Removing the evaluation shard changes the token-count distribution by total variation (TV) 5.156×10^{-5} and Jensen–Shannon (JS) divergence 3.698×10^{-9} nats; the median absolute $\log P(t)$ shift is 1.50×10^{-4}, and only 13 evaluator-compatible subword tokens (0.052%) change decile (Table 34). Even the stricter train-only prior from shards 0–127 remains nearly rank-equivalent to the full prior (Spearman $= 0.999995$) and changes decile for 130 tokens (0.525%). Thus the held-out text is not used for language-model parameter updates, and the contribution of shard 139 to the aggregate unigram table has negligible effect on the frequency ranks and deciles used by FAA.

Why FineWeb-Edu? FineWeb-Edu is filtered toward educational content, so its long tail contains many semantically meaningful entities, scientific terms, domain expressions, and proper nouns, although noisy rare forms still remain. This makes D0 PPL reductions more interpretable than they would be on an unfiltered web crawl, while still requiring the byte-token and occurrence-mass controls reported below.

Table 34: **Unigram-prior sensitivity to shard split.** Metrics compare alternative shard choices with the full shards 0–139 prior and use the evaluator-compatible token set.

Prior estimate	Token share	TV	JS	Spearman	Median $\lvert\Delta \log P\rvert$	Decile changed
no-eval shards 0–138	0.997522	5.156×10^{-5}	3.698×10^{-9}	1.000000	0.000150	13 (0.052%)
train-only shards 0–127	0.961829	5.65×10^{-4}	3.784×10^{-7}	0.999995	0.001458	130 (0.525%)

N.1 Tokenizer construction details

We employed a two-phase BBPE training strategy for robust vocabulary construction:

1. **Singleton exclusion**: Phase 1 excludes strings observed only once within each shard from the stored tokenizer-training state (`min_frequency=2`).
2. **Large candidate buffer**: Phase 2 prunes the accumulated string-count map during state merging when it exceeds $100 \times V_{\text{target}}$ entries.

Quantile-floor probability assignment for zero-count and extreme-tail tokens is described in Appendix B.

Baseline fairness. In the primary Baseline–FAA comparison, Baseline and FAA use the same tokenizer, corpus split, and optimizer settings. The frequency table is shared for decile analysis, and among these two primary models it is supplied only to FAA at training/inference time, so tokenizer differences do not explain the FAA–Baseline gap. Frequency-aware baselines in Appendix H.1 intentionally receive the same corpus statistics at their respective intervention points.

Zipf's law in BBPE tokens. A natural question is whether Zipf's law—originally formulated for word frequencies—is a useful descriptive reference for BBPE token counts. Figure 16 shows that BBPE token frequencies are approximately log-linear on a log–log rank–frequency plot. Fitting $f(r) = C \cdot r^{-s}$ across different rank ranges yields Table 35.

Table 35: Log–log exponent estimates for tokenizer rank–frequency ranges.

Rank range	Exponent s	R^2
10–5,000	0.91	0.997
100–10,000	0.97	0.995
1,000–20,000	1.11	0.998

The fitted exponents, equivalently the negative log–log slopes, fall near the classical Zipf range ($s \approx 1$), with $R^2 > 0.99$ for these log–log fits. We use this as a descriptive check on the tokenizer-induced frequency spectrum, not as a formal power-law test.

Byte-token (alphabet) distribution. BBPE includes 256 single-byte tokens (the initial alphabet). Although they are only $\sim$1% of the vocabulary, they account for 24.8% of non-special token occurrences in the prior-estimation corpus and occupy 22 of the top-100 ranks by frequency. This motivates treating byte tokens separately in analyses (as we do for decile PPL).

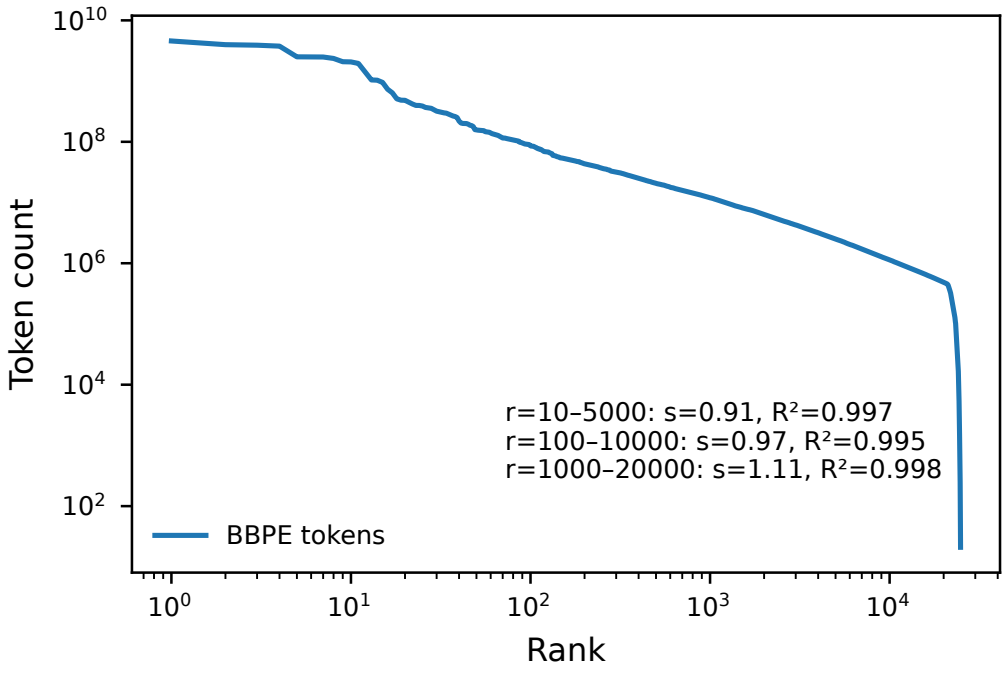


Figure 16: Log–log rank–frequency plot for the 25k BBPE tokenizer trained on FineWeb-Edu 100B. Fitted exponents are reported in Table 35.

O CROSS-ARCHITECTURE AND DOMAIN TRANSFER

GPT-NeoX cross-architecture validation. To verify that FAA generalizes beyond LLaMA-style architectures, we evaluate on a GPT-NeoX-style 150M model (Biderman et al., 2023; Black et al., 2022), which differs from LLaMA in using LayerNorm (vs. RMSNorm), GELU MLP (vs. SwiGLU), and parallel attention-MLP residuals (vs. sequential).

Table 36: GPT-NeoX 150M comparison: mean ± population standard deviation over three paired seeds.

Configuration	PPL	D0 PPL
Baseline	11.45 ± 0.01	54.38 ± 0.21
FAB+IWVS	11.45 ± 0.01	54.13 ± 0.28
Full FAA	**11.17 ± 0.01**	**51.47 ± 0.33**

Table 37: GPT-NeoX downstream accuracy (%): mean ± population standard deviation over the same three seeds as Table 36.

Configuration	ARC-E	ARC-C	HSwag	PIQA	LAMBADA
Baseline	57.23 ± 0.04	23.83 ± 0.67	31.92 ± 0.06	64.91 ± 0.87	31.27 ± 0.29
FAB+IWVS	**57.76 ± 0.75**	**24.83 ± 0.64**	31.84 ± 0.20	64.94 ± 0.38	**31.70 ± 0.51**
Full FAA	57.32 ± 0.79	24.63 ± 0.38	**32.29 ± 0.11**	**65.56 ± 0.27**	31.68 ± 0.60

Tables 36 and 37 compare Baseline, FAB+IWVS, and Full FAA over three paired seeds. Full FAA improves overall PPL and D0 PPL in every seed and improves all five downstream means. FAB+IWVS alone has nearly unchanged mean overall PPL and a modest D0 reduction, while

improving four downstream means. The component gains therefore depend on architecture, whereas the Full FAA gains transfer to this NeoX setting.

Specialized-domain evaluation without prior recomputation. To test whether FAA remains useful when the evaluation distribution differs from the pre-training corpus, we evaluate the pre-trained 150M Baseline and Full FAA checkpoints on PubMed biomedical articles from the long-document summarization corpus of Cohan et al. (2018) *without any further training*. Both models keep the original 25k tokenizer from FineWeb-Edu pre-training, and FAA keeps the original FineWeb-Edu-derived unigram table rather than recomputing $I(t)$ on the target corpus. This therefore probes domain shift under moderate prior mismatch rather than full domain adaptation. The held-out domain corpus contains 52.5M PubMed tokens under the same tokenizer.

Table 38: Zero-shot PubMed evaluation without prior recomputation. FAA keeps the FineWeb-Edu unigram prior, and D0/D9 use FineWeb-Edu deciles.

Domain	PPL (Baseline)	PPL (FAA)	D0 (Baseline)	D0 (FAA)	D9 (Baseline)	D9 (FAA)
PubMed	11.76	11.10	80.49	68.65	13.35	12.88

Table 38 reports a zero-shot PubMed summary with overall PPL and D0/D9 rare/common deciles: FAA lowers overall PPL and reduces D0 PPL by 14.7% (80.49 $\rightarrow$ 68.65) while reducing D9 PPL by 3.5% (13.35 $\rightarrow$ 12.88). These results are not full domain adaptation, but they suggest that FAA's gains can survive specialized-domain shift even before target-domain prior recomputation, motivating the per-source CPT setting detailed in Appendix P.

P Continued pre-training details and full results

Table 39 below gives the full per-benchmark CPT results. This appendix details the dataset, hyperparameters, and biomedical benchmark definitions.

Table 39: **PubMed CPT results (150M, 50k steps).** PPL uses the FineWeb-Edu held-out shard; benchmark columns group general and biomedical tasks.

	PPL↓	General					Biomedical		
Model		ARC-E	ARC-C	HSwag	PIQA	LAMBADA	PubMed RCT	ChemProt	MedMCQA
Baseline (no FAA)	11.17	57.03	25.00	32.19	64.53	31.88	10.49	3.45	**32.03**
FAA + FineWeb-Edu	**10.66**	**59.85**	**26.28**	**33.31**	**65.51**	**33.59**	12.10	**4.20**	31.80
FAA + per-source	**10.66**	59.51	26.02	33.19	**65.51**	33.32	**13.84**	4.11	31.82

Dataset. We use the full 2026 National Center for Biotechnology Information (NCBI) PubMed baseline distribution.[3] For each `PubmedArticle` record, we extract the `ArticleTitle` and every `AbstractText` element (labels, when present, are prepended as "`LABEL: text`"), and concatenate title and abstract with a blank-line separator; records shorter than 200 characters are discarded. We apply no deduplication, language filter, or update-file supplementation. The resulting documents are written to 50,000-document Apache Arrow shards; the final NCBI baseline Extensible Markup Language (XML) file is reserved for evaluation and the remaining files constitute the training split. Raw XML archives are MD5-verified against NCBI checksums, and Arrow shards are recounted for row-level integrity. All tokenization uses the same 25k BBPE tokenizer from pre-training. Training follows an 80/20 document-level mix of PubMed and FineWeb-Edu replay to prevent catastrophic forgetting of general-domain capabilities. The per-source PubMed prior is computed from a uniform 5% document sample of the PubMed training split using the same pre-training tokenizer and the same p=0.5% quantile-floor clamping rule as the base FineWeb-Edu prior; the numeric floor is recomputed from the PubMed sample. Both frequency tables (PubMed and FineWeb-Edu replay) are stored as 1D tensors indexed by token ID with $O(1)$ lookup, and the active prior is selected per document based on its source.

[3] NCBI PubMed baseline FTP index.

Hyperparameters. All three conditions (Baseline without FAA, FAA + FineWeb-Edu, and FAA + per-source) use identical CPT settings: peak learning rate 6×10^{-5} (1/10× the pre-training peak), minimum learning rate 6×10^{-6}, 500-step linear warmup with cosine decay, 50,000 total steps, and the same optimizer settings as pre-training (AdamW, β_1=0.9, β_2=0.95, weight decay 0.1, gradient clipping 1.0; see Appendix D). Checkpoints are saved every 5,000 steps; final checkpoints are used for all evaluations.

Evaluation protocol. The FineWeb-Edu held-out shard is used as the general-domain evaluation corpus for all three CPT conditions. The reserved PubMed evaluation shards are not mixed into CPT training; they are used only for PubMed-domain checks outside the general-domain PPL evaluation. In FAA + per-source, `tokenizer_path` stores the *domain* (PubMed) prior and `replay_tokenizer_path` stores the *general* (FineWeb-Edu) prior. During FineWeb-Edu evaluation, we therefore feed the replay tokenizer's log-probs to the model (auto-detected when `eval_data_dir` matches `replay_data_dir`; overridable via a `--eval-tokenizer` command-line interface (CLI) flag), matching the source-conditioned prior used during training. Baseline and FAA + FineWeb-Edu are unaffected since they do not set a separate replay tokenizer.

Biomedical benchmarks. In addition to the five general benchmarks used throughout the paper, we evaluate on three biomedical benchmarks:

- **PubMed RCT** (Dernoncourt & Lee, 2017) (5-way): Sentence-role classification in PubMed abstracts (Background, Objective, Methods, Results, Conclusions). We use the 20k subset test split and evaluate zero-shot accuracy.
- **ChemProt** (Krallinger et al., 2017) (chemical–protein relation extraction): Chemical–protein interaction types from PubMed abstracts. We evaluate zero-shot accuracy on the test split using our 11-choice offline prompt.
- **MedMCQA** (Pal et al., 2022) (4-choice; chance =25%): Medical entrance examination questions. Zero-shot accuracy on the validation split (4,183 examples; test labels are hidden).

Q GLOBALLY SHARED LEARNED-TOKEN CONTROLS

Matched shape and consumers. The static prior assigns one corpus self-information value to each vocabulary item, shared by all layers and heads. To isolate the effect of learning the token values while preserving this sharing structure, we train a single unconstrained vector $b \in \mathbb{R}^{25,024}$ and retain the same 144 per-layer/head coefficients:

$$s_{\ell,h,i,j} = \frac{\mathbf{q}_{\ell,h,i}^{\top}\mathbf{k}_{\ell,h,j}}{\sqrt{d_k}} + \alpha_{\ell,h} b[t_j], \qquad r_b(t) = \frac{b[t] - I_{\min}^{(0)}}{I_{\max}^{(0)} - I_{\min}^{(0)} + 10^{-8}}. \tag{13}$$

IWVS and FAIG consume the same live r_b, replacing their fixed token-indexed inputs; the content-dependent IG remains unchanged. Thus the control has one shared token vector, not an independent $L \times H \times |\mathcal{V}|$ bias table. This tests fixed versus learned token conditioning throughout Full FAA while preserving the table's global sharing and the three consumer architectures. Total parameters increase from 159,285,036 to 159,310,060, exactly 25,024. The five structural special tokens remain excluded from the learned signal, with zero score bias, the neutral response coordinate, and IWVS scale one; the 25,019 eligible entries include all 256 byte tokens.

Initialization and normalization. We use corpus self-information, the constant midpoint $(I_{\min}^{(0)} + I_{\max}^{(0)})/2$, or a seed-specific permutation of the corpus self-information values as three initializers. Each is trained with three seeds using the Full FAA recipe: 197,175 optimizer steps, global batch 256, and sequence length 2,048. The resulting nominal training budget is 103,376,486,400 target positions. Both b and α have zero weight decay; α starts at zero. Corpus initialization starts from the same FAA function as the static configuration, then permits b to change. Midpoint initialization removes token-wise distinctions at initialization, whereas shuffling preserves the vocabulary-level multiset but destroys its alignment.

The scalar anchors $I_{\min}^{(0)} = 3.2068865$ and $I_{\max}^{(0)} = 20.0394421$ stay fixed. We neither recompute live extrema nor clip r_b; the vector is free to move outside its original coordinate range. These

anchors remain corpus-derived scalar metadata, so the control removes the fixed *per-token* prior, not every corpus-derived quantity. The learned vector is a feature table, not a normalized probability distribution. IWVS uses the same amplification range $(1, 10)$ as static FAA.

Held-out and downstream performance. Table 40 reports PPL and all five zero-shot downstream accuracies as three-seed means with population standard deviations. Corpus-initialized learned FAA is close to static FAA in overall PPL, and both have slightly lower mean PPL than the midpoint- and shuffled-initialized controls. Together with the downstream comparisons below, these results show a modest overall advantage for corpus-informed conditioning, attained without extra token-wise learned parameters by static FAA. Their learned token orderings nevertheless differ, as quantified below.

Table 40: Shared-token PPL and zero-shot downstream accuracy (%): mean ± population standard deviation across three seeds ($n = 3$ denominator), computed before display rounding. Static uses a fixed corpus table; other rows learn the table from the indicated initialization. All checkpoints use 197,175 optimizer steps; PPL evaluation scores 289,013,760 target positions. Task splits and scoring follow Appendix N. Bold marks the best displayed mean for each metric, including ties.

Prior	PPL↓	ARC-E	ARC-C	HSwag	PIQA	LAMBADA
Static	**10.80 ± 0.01**	**58.85 ± 0.54**	**25.86 ± 0.53**	33.27 ± 0.05	66.18 ± 0.47	**33.50 ± 0.49**
Corpus	**10.80 ± 0.01**	58.78 ± 1.03	24.72 ± 0.22	**33.30 ± 0.05**	65.94 ± 0.19	33.42 ± 0.72
Midpoint	10.81 ± 0.01	58.68 ± 0.72	24.80 ± 0.28	33.25 ± 0.30	**66.32 ± 0.09**	33.18 ± 0.70
Shuffled	10.81 ± 0.01	58.31 ± 0.67	25.00 ± 1.16	33.05 ± 0.18	65.80 ± 0.62	32.96 ± 0.47

Static FAA has higher means on four of five tasks than the corpus-initialized control and on all five than the shuffled-initialized control; midpoint initialization leads on PIQA, and corpus initialization on HellaSwag. Corpus initialization also exceeds midpoint on three of five downstream means and shuffled on four of five. Thus the reported means generally favor static and corpus-initialized conditioning over the frequency-unaligned initializers, with small, task-dependent differences.

What changes in the learned vector? Table 41 uses the same 25,019 eligible tokens, including byte tokens, for all nine runs. Corpus initialization preserves strong frequency structure: final Pearson correlation is 0.9819–0.9827, and a centered linear projection onto corpus self-information explains 96.52% of the final vector's centered energy on average. The update is not zero: its root-mean-square (RMS) magnitude is 0.425–0.439 in the raw scalar coordinate, versus an initial token-wise standard deviation of 1.787. The final vectors retain strong frequency structure while adjusting individual token values. Across seeds, Spearman rank correlations lie in $[0.9347, 0.9373]$ for corpus initialization, $[-0.0506, -0.0305]$ for midpoint initialization, and $[-0.0088, 0.0579]$ for shuffled initialization.

Table 41: Learned-token geometry: mean ± population standard deviation of per-run statistics across three seeds ($n = 3$ denominator), computed before display rounding. Each run uses the same 25,019 eligible entries, including byte tokens. Correlations compare the final vector to corpus self-information. Update RMS is for $b_{\text{final}} - b_{\text{init}}$ in the raw scalar coordinate, not a probability change. These are descriptive statistics, not quality rankings.

Initialization	Pearson r	Spearman	Update RMS
Corpus-init	0.9824 ± 0.0004	0.9360 ± 0.0010	0.431 ± 0.006
Midpoint-init	-0.0976 ± 0.0072	-0.0425 ± 0.0087	0.430 ± 0.011
Shuffled-init	0.0046 ± 0.0205	0.0147 ± 0.0306	0.537 ± 0.011

Training dynamics of frequency conditioning. Figure 17 separates two comparisons within Full FAA. With fixed priors, the aligned corpus table yields lower training loss than random or shuffled tables after 5k steps, although this ordering is reversed in the first interval. With learnable token tables, static FAA and corpus initialization remain close; differences from midpoint and shuffled initialization are much smaller than the fixed-prior differences. In late training, both corpus-informed configurations have slightly lower loss than the frequency-unaligned initializers. This complements the FAA–Baseline learning curves in Figure 3, which also vary the content gate.

For each paired run we integrate the difference in rank-0 local training loss over the shared observed step interval and divide by its length.

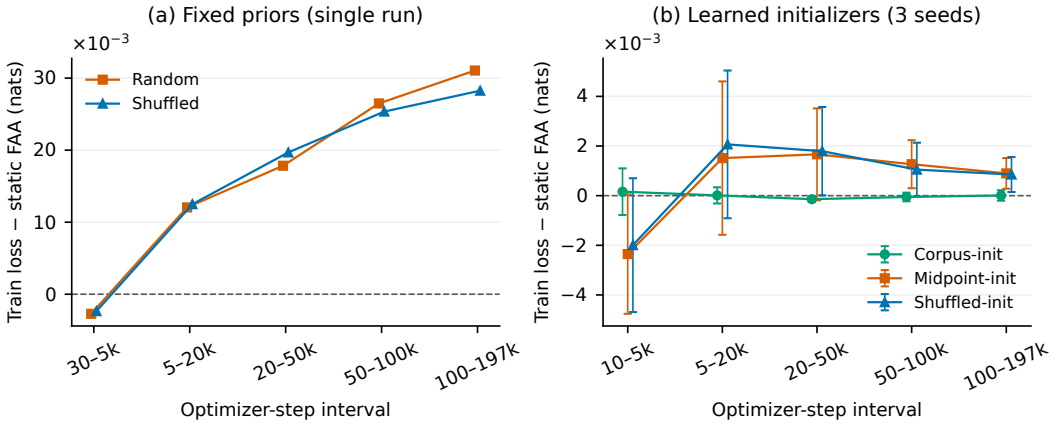


Figure 17: Training-loss differences relative to static corpus FAA, averaged within each optimizer-step interval. Positive values favor static FAA. (a) Fixed random and shuffled priors, a single run. (b) Learned initializers, mean ± population standard deviation across three paired seeds. Panels use different vertical scales. Points are interval averages, not instantaneous losses; lines connect interval summaries. The observed spans are steps 30–197,170 in (a) and 10–197,170 in (b), with no extrapolation to step zero.

R Frequency pathways with the gate held fixed

We cross FAB+IWVS (enabled or disabled) with the content gate IG (enabled or disabled), giving a 2×2 component comparison at 150M. Table 42 reports the complete target-token decile PPL on the held-out set. Holding IG fixed, adding FAB+IWVS reduces D0 PPL by 5.9% and D9 by 2.6%. Without IG, the corresponding reductions are 7.2% and 2.8%. Adding IG to FAB+IWVS reduces D0 by 3.5% and D9 by 2.0%. Both pathways help, while the frequency pathways have a larger rare-to-common differential in this matched 150M diagnostic. This is a comparison of components enabled during training.

Table 42: Gate-matched target-token PPL across all ten deciles (150M, single run; full 289.0M-target held-out evaluation). D0 is rarest, D9 most common. The comparison crosses FAB+IWVS and the content gate IG, each enabled or disabled during training.

Decile	Baseline	IG-only	FAB+IWVS	FAB+IWVS+IG
D0	53.79	51.17	49.90	**48.15**
D1	51.42	49.13	47.65	**46.32**
D2	66.53	63.66	61.68	**59.91**
D3	65.20	62.22	60.43	**58.62**
D4	63.59	60.84	59.07	**57.23**
D5	60.84	58.13	56.41	**54.81**
D6	56.89	54.58	53.00	**51.45**
D7	46.82	44.90	43.71	**42.52**
D8	39.99	38.55	37.56	**36.56**
D9	13.17	12.88	12.80	**12.54**

S Attention diagnostics

S.1 Pooled attention

The pooled diagnostic in Table 5 averages received attention over 65,536 held-out token occurrences per checkpoint. D0 received attention increases by 2.41× and D9 by 1.56× in the three-seed means.

Thus, FAA redistributes attention toward rare keys relative to frequent keys, while increasing the attention received by both groups. For each pair, the separate 2,000-resample batch-block 95% intervals do not overlap for either decile.

T LIMITATIONS, BROADER IMPACTS, AND FUTURE WORK

Limitations. FAA's prior is unigram and tokenizer-indexed: it does not model context-conditional surprisal, multi-token concepts, or domain-specific shifts unless the prior is recomputed or source-conditioned. Prior analyses identify token-frequency neurons that shift next-token distributions toward or away from the unigram marginal (Stolfo et al., 2024); FAA's external static prior intervenes inside attention rather than at the output distribution, and cannot by itself distinguish globally rare tokens from locally predictable ones. Because rarity can also reflect noise rather than meaningful domain terms, corpus and tokenizer diagnostics remain important before deployment. We have not yet measured calibration error, memorization/extraction risk for rare strings, multilingual and low-resource behavior, language-model quality with much larger vocabularies such as 100k-token production tokenizers, or post-training embedding anisotropy (Gao et al., 2019; Yu et al., 2022), which our introduction cites as motivation but does not directly verify. SentencePiece unigram and WordPiece quality are also untested. Multi-seed evidence covers the primary 150M (3 seeds) and 450M (2 seeds) comparisons and the 150M learned-token and NeoX comparisons; 1.2B and the fixed-prior controls remain single-seed. Learning a shared token vector can achieve similar overall PPL, so the evidence supports the usefulness and economy of a static prior rather than universal superiority over learned conditioning.

Broader impacts. Improved rare-token modeling may support factual accuracy in scientific and medical applications where long-tail terminology is central to interpretation, consistent with our PubMed CPT gains. It also lowers the modeled FLOPs needed to reach a target loss (Appendix E); actual energy savings require end-to-end energy measurements and are not measured here. The same mechanism reflects token-level statistics in the training corpus: because rare tokens can correspond to private identifiers, noisy artifacts, or dialectal variation, FAA's amplification effect should not be assumed uniformly beneficial. We recommend pairing FAA with corpus audits, tokenizer diagnostics, and—where applicable—privacy/extraction checks before deployment.

Future work. Because FAA's prior is a non-parametric, tokenizer-indexed lookup table decoupled from the learned weights, it can be recomputed or swapped at inference time. This opens two natural extensions beyond the CPT study summarized in Section 4.3 and detailed in Appendix P. First, *test-time adaptation*: unlike nearest-neighbor test-time training, which updates model parameters on retrieved examples (Hardt & Sun, 2024), FAA could adapt only the unigram prior by replacing it with a target-domain estimate. Second, *streaming-corpus adaptation*: the model could update $I(t)$ from streaming corpora or accepted generation traces, connecting FAA to self-training (Zelikman et al., 2022), self-refinement (Madaan et al., 2023), and self-adaptive language-model frameworks (Tao et al., 2024; Zweiger et al., 2025). Relative to this broader family, FAA exposes a *weight-preserving* adaptation pathway: swapping the tokenizer-indexed unigram prior reshapes attention allocation without modifying learned weights, which makes prior swaps cheaper to revert than weight-updating variants. Future evaluations should add document-frequency/IDF-style priors, token-importance schedules, calibration metrics, privacy/extraction tests, 100k+ vocabulary tokenizers, and multilingual or low-resource tokenizers. Streaming adaptation in any of these forms still requires safeguards against feedback loops and noisy-tail effects.