PRETRAINING TRANSFORMERS WITH QUANTIZED SOFTMAX IN ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-precision Transformer systems increasingly quantize attention matrix multiplications, while softmax often remains at higher precision. During pretraining, an approximate softmax changes the gradients that train the model as well as its forward computation. We study this interaction with K-interval attention, which approximates the exponential using $K+1$ grid values. We vary per-row grid calibration, interpolation versus hard rounding, and the placement of a straight-through surrogate relative to normalization. We derive the corresponding backward rules, including calibration derivatives, and compare these choices in pretraining experiments matched on model, data, and optimizer. Detaching the row extrema leaves the forward computation unchanged but produces a delayed increase in validation loss. With hard rounding at $K{=}4$, min–max calibration and a pre-normalization surrogate incur a large loss gap; changing either choice substantially reduces it. At 124M parameters and 2.5B training tokens, fixed-window calibration with a post-normalization surrogate yields a validation loss gap of $+0.019$ nats relative to softmax at $K{=}4$; with a pre-normalization surrogate yields $+0.004$ nats at $K{=}16$.

1 INTRODUCTION

Low-precision Transformer training has moved the linear projections and, increasingly, the attention matrix products to narrow formats, while the exponential and its row reduction are kept in FP16/FP32 by design (Xiao et al., 2023; Mishra et al., 2025; Hernández-Cano et al., 2025; Zhang et al., 2026; Ding et al., 2026). Whether a low-precision reconstruction of the exponential can be used *during pretraining* is left open by several works (Shkolnik et al., 2024; Zhang et al., 2025; Mishra et al., 2025; Ye et al., 2026).

The question differs in kind from the inference-time one. At inference an approximate operator perturbs a fixed function; during training it defines the learning rule, because the model sees the approximate forward *and* is updated by whatever gradient the implementation attaches to it. For softmax, detaching the row maximum is harmless: shift invariance makes its gradient cancel exactly, so subtracting the maximum is treated as a numerical convenience rather than part of the function. The same habit carried over to a calibrated grid is not harmless, because there the row extrema also set the grid; quantization-aware training frameworks, which simulate quantization in floating point ("fake quantization"), likewise keep their observer statistics off the autograd tape (PyTorch contributors, 2026).

We study a family simple enough that every forward and backward term is closed-form (Figure 1, Table 1). Three design axes define an operator: *calibration*, row-wise min–max calibration (MinMax) over each row's own extrema or a fixed window of τ nats anchored at the row maximum with zero weight below it (FWM); *reconstruction* of the exponential on K intervals from $K+1$ tabulated grid values, by piecewise-linear interpolation (LERP) or by rounding each score to the nearest grid center (Nearest); and *surrogate placement*, Weight-STE on the unnormalized weights before normalization or Prob-STE on the normalized probabilities. We call the family K-interval attention. The experiments study the training properties of coarse reconstruction in fp32 attention arithmetic (TF32 matmuls in training, §4); we measure no kernels and claim no speed-up.

- **The same forward can fail solely because the calibration backward is incomplete.** Detaching the row extrema leaves the forward unchanged but drops two Jacobian terms and violates the zero-sum identity that shift invariance implies. Training then tracks softmax for 25–30M tokens,

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.

diverges, and ends 0.65–3.07 nats above the matched run with full calibration gradients (the backward that keeps the derivatives through the row extrema; replicated on two further seeds). Restoring the zero-sum identity alone does not repair it (§5.1).

- **For hard reconstruction, calibration policy and surrogate placement strongly affect training quality and stability.** MinMax×Weight-STE at $K{=}4$ ends $+0.39$ nats above softmax at 250M tokens and $+0.89$ at 2.5B. A post-normalization surrogate or FWM each removes most of the deficit, and the two together add little more. At $K{=}4$ the effects of calibration, of surrogate placement and of their interaction keep their direction at 1B parameters and on all five seeds, and the surrogate effect shrinks with K (§5.2).
- **Coarse deterministic reconstruction need not prevent near-baseline performance.** LERP at the tested $K \in \{4, 16, 32\}$ ends within 0.005 nats of softmax at 2.5B tokens, and FWM–Weight at $K{=}16$ at $+0.004$. Downstream, every large-gap condition scores below softmax on every benchmark configuration, while conditions within 0.01 nats show small, task-dependent differences in both directions (§5.3, §5.5).

That both failures reflect a gradient defect which grows with the learned score geometry is a hypothesis; §5.4 states it with its counter-examples.

2 Related work

Low-precision Transformers and the softmax path. Weight and activation quantization and FP8/FP4 training (Frantar et al., 2023; Xiao et al., 2023; Peng et al., 2023; Mishra et al., 2025; Ding et al., 2026) quantize linear layers, GEMM operands and data formats; the attention matmuls are quantized during training by FP8 dot-product attention, SageBwd and Full-Stack FP4 (Hernández-Cano et al., 2025; Zhang et al., 2026; Ding et al., 2026), while the exponential and its normalization stay in high precision, and NVIDIA's MXFP8 recipe leaves reducing their precision "to future work". The nearest inference-side operators at the softmax itself subtract the row maximum and read the exponential from a small lookup table over a window below it. EXAQ (Shkolnik et al., 2024) sets the window width from calibration statistics of the softmax input and clamps inputs below the window to its edge (2–3-bit lookup); IndexSoftmax (Zhong et al., 2026) uses a fixed window, zeroes inputs beyond it, and quantizes table entries and probabilities to UINT8. EXAQ and BAPS (Ye et al., 2026) state that training is untested, and IndexSoftmax is designed as a training-free drop-in replacement. Our scope is the normalization operator itself, the derivative of its data-dependent row calibration, and where a surrogate sits relative to normalization (Appendix H).

Approximate and non-exponential softmax. Softermax (Stevens et al., 2021) and I-BERT (Kim et al., 2021) approximate the exponential with static scales and fine-tune downstream; ConSmax (Liu et al., 2024) removes the row reduction, keeps the exponential, and learns per-head parameters. Zhang et al. (2022) argue trainability of a base-2 classifier softmax from gradient structure (the Jacobian changes by a constant $\ln 2$); under per-row calibrated reconstruction the Jacobian terms are neither constant nor absorbable into the learning rate. Zhang et al. (2023) find that i.i.d. softmax error above 10^{-6} breaks training even under an exact backward.

Non-exponential attention is trainable with a stabilizer in each case (Wortsman et al., 2023; Shen et al., 2023; Zhang et al., 2021; Ramapuram et al., 2024; Katharopoulos et al., 2020); our LERP at $K{=}1$ is a single-interval affine reconstruction; no linear-complexity benefit is claimed.

Surrogates and calibration gradients. The straight-through estimator was named and analysed by Bengio et al. (2013); binarized networks (Hubara et al., 2016) needed clipped surrogates and explicit weight clipping because a hard forward is insensitive to latent magnitude, and Yin et al. (2019) show that a surrogate must match the hard forward at its extremes for the coarse gradient to correlate with the true one. Straight-through Gumbel-softmax (Jang et al., 2017; Maddison et al., 2017) is the probability-level STE with only one place to sit, because the relaxation is the normalization; our reconstruct-then-normalize operator creates the weight-level/probability-level choice. PACT, LSQ and TQT (Choi et al., 2018; Esser et al., 2020; Jain et al., 2020) learn clipping or step-size *parameters* with a gradient, whereas our calibration quantities are per-row statistics of the scores: LSQ's step-size gradient is the per-layer analogue of our per-row $\partial w/\partial M$, $\partial w/\partial m$, but dropping it freezes a parameter, whereas dropping ours changes $\partial L/\partial s$ itself. ITA (Islamoglu et al., 2023) sets the clipping range of an integer, shift-based attention softmax by quantization-aware training; the quantity it learns is a quantizer scale, not a per-row statistic of the scores. We are not aware of prior work that back-propagates through a per-row calibration statistic in attention pretraining.

3 K-INTERVAL ATTENTION AND ITS GRADIENTS

Objects and calibration. For one causal row of fp32 scores $s = (s_1, \ldots, s_n)$, $S = QK^\top/\sqrt{d_h}$, every operator produces weights $w_j \geq 0$, normalizes them to $P_j = w_j/W$ and outputs $O = PV$ (notation: Table 1). *MinMax* places the grid on the observed row range: $h = (M - m)/K$, $b_r = (1 - r/K)(m - M)$ for $r = 0, \ldots, K$, key j in interval r_j at local position $t_j \in [0, 1]$; all $K+1$ grid values e^{b_r} move with the row extrema, and h is set by the row *minimum*, so the learned negative tail sets the resolution near the row maximum, where the high-weight keys are. *Fixed Window to Max* (FWM) anchors a fixed-width window $[M - \tau, M]$ at the row maximum: $h = \tau/K$, $b_r = -\tau + rh$, $\tilde{z}_j = \mathrm{clip}(z_j, -\tau, 0)$, and, by explicit implementation choice, $w_j = 0$ for $z_j < -\tau$ (zero tail); it needs only M, whose key is always in the window, and keeps the gradient through M. $\tau = 6$ was fixed before any FWM training run by a multi-stage inference-time screen on frozen softmax-trained weights (Appendix C); no other τ was trained. Edge cases (degenerate rows, ties, index switches, the window boundary) are specified in Appendix A; the formulas below assume a nondegenerate row, a locally fixed interval index and unique extrema where they enter.

Reconstruction. LERP interpolates between adjacent exponential grid values, $w_j^L = (1 - t_j)e^{b_{r_j}} + t_j e^{b_{r_j+1}}$, a piecewise-linear reconstruction of the exponential whose weights are continuous rather than grid values. Nearest rounds each score to the nearest grid *center* in score coordinates (not the nearest exponential value), $\ell_j = r_j + \mathbb{1}[t_j \geq \frac{1}{2}]$, $w_j^N = e^{b_{\ell_j}}$: grid-centered rounding bins with boundaries halfway between grid points (Figure 1a,b), so distinct scores in one bin get the same unnormalized weight. Nearest is "hard" in the sense that it selects one grid value per key; attention itself is not one-hot. The *index* ℓ_j is piecewise constant with zero derivative almost everywhere.

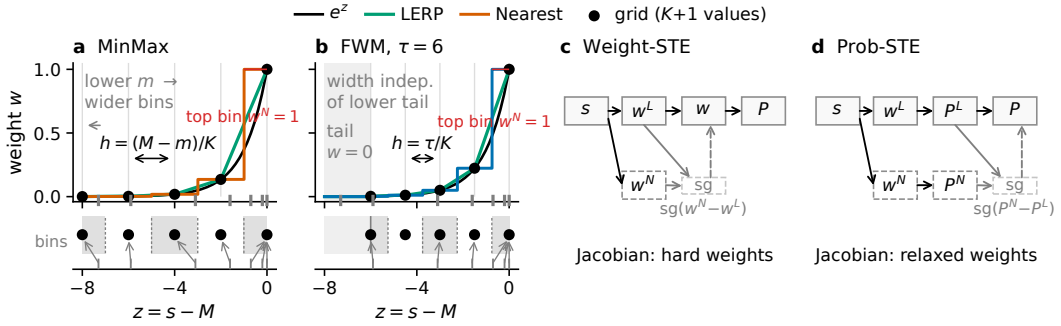


Figure 1: **The operator family** (K=4; three design axes). (a) MinMax calibrates the $K+1$ exponential grid values (dots) over the observed row range, $h = (M - m)/K$; a lower minimum widens every bin at fixed M, K. (b) FWM anchors a fixed-width window at the row maximum, $h = \tau/K$, zero weight below it (shaded tail). LERP (green) interpolates between adjacent grid values; Nearest (step) rounds to the nearest grid center. Strips: Nearest rounding bins (boundaries halfway between grid points; scores in one bin share w^N). (c, d) Weight-STE applies the straight-through replacement to the unnormalized weights, Prob-STE to the normalized probabilities: the node *sg* (stop-gradient: identity forward, zero derivative backward) receives both reconstructions (c) or both normalized probabilities (d), and its output is added to the LERP branch (dashed arrow); the hard forward $P = P^N$ is the same, the normalization Jacobian is evaluated at the hard (c) or relaxed (d) weights.

Table 1: Method identity (three design axes) and notation; edge cases in Appendix A.

Name	Calibration	Reconstr.	Backward
softmax	—	exp	autograd
LERP	MinMax	LERP	exact Jacobian (2)
MinMax–Weight	MinMax	Nearest	Weight-STE (3)
MinMax–Prob	MinMax	Nearest	Prob-STE (4)
FWM–Weight	FWM, τ=6	Nearest	Weight-STE
FWM–Prob	FWM, τ=6	Nearest	Prob-STE

s_j: score of key j in a causal row; M, m: row max, min; $z_j = s_j - M \leq 0$; K: intervals; h: interval width, $(M - m)/K$ (MinMax) or τ/K (FWM); τ: window width, 6 nats; b_r: grid points in z; w_j: unnormalized weight (w^L interpolated, w^N rounded); $W = \sum_l w_l$; $P_j = w_j/W$; $g_j = \partial L/\partial P_j$; $\gamma_j = \partial L/\partial w_j$; Δ_j: slope of w_j^L in its interval; $\mathrm{sg}(x)$: stop-gradient, x forward, zero derivative backward; ΔNLL: validation NLL minus softmax, nats/token, positive when worse.

Under MinMax the grid values move with the extrema, so the hard map still has extremum-mediated derivatives where the index is locally constant (Appendix B); these carry no sensitivity to the index selection, which the surrogate below supplies. Under FWM the hard weights are locally constant, and the zero tail is discontinuous at $z = -\tau$.

The full calibration gradient and the zero-sum identity. With $g_j = \partial L/\partial P_j$ the normalization Jacobian gives $\gamma_j \equiv \partial L/\partial w_j = (g_j - \sum_i P_i g_i)/W$, and inside an interval the LERP slope is the secant $\Delta_j = (e^{b_{r_j+1}} - e^{b_{r_j}})/h$. Because the MinMax grid depends on M and m, so does w_j^L; with $\rho_j = (r_j + t_j)/K$ and $X_j = \frac{1}{K}[(1-t_j)r_j e^{b_{r_j}} + t_j(r_j{+}1)e^{b_{r_j+1}}]$,

$$\frac{\partial w_j^L}{\partial M} = -w_j^L - \Delta_j \rho_j + X_j, \qquad \frac{\partial w_j^L}{\partial m} = -\Delta_j(1-\rho_j) + w_j^L - X_j, \tag{1}$$

$$\frac{\partial L}{\partial s_i} = \gamma_i \Delta_i + \mathbb{1}[i{=}\arg\max] \sum_j \gamma_j \frac{\partial w_j}{\partial M} + \mathbb{1}[i{=}\arg\min] \sum_j \gamma_j \frac{\partial w_j}{\partial m}. \tag{2}$$

The three per-key terms sum to zero, hence $\sum_i \partial L/\partial s_i = 0$, the gradient identity that shift invariance $F(s + c\mathbf{1}) = F(s)$ implies; under FWM the identity holds with $\partial w_j/\partial M = -\Delta_j$ and no m term. We call (2) the *full calibration gradient*. For the differentiable LERP operator it is the exact Jacobian, calibration included; for the straight-through rules below it is the set of calibration derivatives that the surrogate carries, not the derivative of the hard forward. The *detach* variant keeps only $\gamma_i \Delta_i$ and violates the identity (§5.1). For a hard forward the index path's derivative is zero almost everywhere, so we adopt a LERP surrogate, and the question becomes which calibration derivatives the surrogate carries.

Two places to put the surrogate. Weight-STE applies the straight-through replacement to the unnormalized attention weights before sum normalization, Prob-STE to the normalized attention probabilities; $\mathrm{sg}(x)$ returns x in the forward pass and has zero derivative in the backward pass:

$$\text{Weight-STE: } w = w^L + \mathrm{sg}(w^N - w^L),\ P = \tfrac{w}{\sum_l w_l}, \qquad \gamma_j^W = \frac{g_j - \langle g \rangle_{P^N}}{W^N}, \tag{3}$$

$$\text{Prob-STE: } P = P^L + \mathrm{sg}(P^N - P^L),\ P^L = \tfrac{w^L}{W^L}, \qquad \gamma_j^P = \frac{g_j - \langle g \rangle_{P^L}}{W^L}. \tag{4}$$

Both rules define the same mathematical hard forward $P = P^N$ (the two floating-point implementations differ by rounding, measured on frozen checkpoints in Appendix C.2). Both use the same LERP reconstruction derivatives Δ, $\partial w/\partial M$, $\partial w/\partial m$ of (2), because both stop the gradient on the difference branch. They differ in the point at which the normalization Jacobian is evaluated, so the centering distribution in the numerator and the normalization factor can both differ. Prob-STE's backward is the Jacobian of $s \mapsto \mathrm{normalize}(w^L(s))$ contracted with the hard-forward upstream gradient. Weight-STE combines the normalization Jacobian at the hard point (W^N, P^N) with the LERP reconstruction derivatives and generally differs from that Jacobian. The two rules coincide when $w^N = w^L$. Their difference $\gamma^W - \gamma^P$ also depends on the within-interval positions of the scores, on g and on the reconstructed weights, so it is not a function of h alone; but a wider interval permits a larger discrepancy between hard and interpolated reconstructions and can therefore amplify it. Under MinMax the width grows with the learned span; under FWM $h = \tau/K$ regardless of the tail. At small h the two surrogates should therefore be hard to distinguish, and the surrogate effect should shrink with K faster under FWM than under MinMax, a local, qualitative expectation that §5.2 tests.

4 Experimental protocol

Training. GPT-2-style models (124M: 12 layers, d=768; 1B: 32 layers, d=1536). Everything outside attention runs under bf16 autocast; the attention operator runs in fp32 with TF32 matmuls enabled in training and disabled at evaluation. Per-suite hardware and software stacks: Appendix C.

Suites (Table 2). Suite B is a balanced $2{\times}2{\times}2$ design (calibration $\times$ surrogate $\times$ $K \in \{4, 16\}$) that estimates the main effects and interactions (the comparison between the two surrogates carries a forward-rounding confound; Appendix C.2). Suite A trains $10\times$ longer (2.5B tokens) under its own cosine schedule, so it tests whether the gaps of suite B persist at a much larger training budget; it is

also the only suite trained long enough for downstream evaluation to be meaningful. Suite C asks whether the ordering reappears at $8\times$ the parameters in an early regime (0.1 tokens/parameter) on another corpus (a probe, not a scale experiment). Suite D holds the full 5×5 grid fixed and varies the seed, the only suite that measures seed-to-seed spread.

Evaluation. Every checkpoint is evaluated on the full validation split of its corpus (976 blocks FineWebEdu, 243 WikiText-103; 1024-token blocks, batch 1, fp32 attention kernel, TF32 off) through the run's *own frozen training code*, since a shared implementation or a larger batch perturbs single Nearest blocks by up to 10^{-2} (Appendix C). All comparisons are paired differences $\Delta\mathrm{NLL}$ in nats per token (GPT-2 BPE), positive when the condition is worse than softmax; δ nats is a perplexity ratio e^{δ} (0.01 nats $\approx 1\%$, $+0.89$ nats $\approx 2.4\times$).

Statistics. A contrast (a paired difference between two conditions, or a difference of such differences) is computed per block and averaged; single-seed suites report 95% circular moving-block bootstrap intervals over validation blocks (block 16, 4000 resamples), covering *evaluation-set sampling only*. Suite D pairs each condition with softmax on the same seed and reports the mean of five paired differences, their seed SD and a 95% t interval (4 df), covering seed-to-seed variation. A single-seed contrast is *separated* at a checkpoint when its CI lower bound exceeds $+0.01$ nats, a pre-declared caution margin, not a significance or equivalence threshold. On five seeds the seed-to-seed spread of a condition's $\Delta\mathrm{NLL}$ exceeds its paired evaluation SE several-fold (§5.2), so single-seed differences below about 0.01 nats are not interpreted, and single-seed intervals never stand in for seed variation. Trajectories, geometry and cross-condition associations are observations; mechanism statements are hypotheses that no intervention in this paper identifies.

Geometry probe. At every checkpoint we recompute the fp32 scores of each attention layer on 8 validation blocks and record, per layer, the row span $M - m$, native width h, query/key norms, softmax entropy, and the per-row total variation TV between native and softmax probabilities on the same model's scores: the operator's distortion of the learned scores, not a training loss or a mediator.

5 Results

5.1 Calibration-gradient intervention: same forward, delayed failure

Before the formal suites the operator was trained with a backward that detached the row extrema: the forward code path is unchanged apart from the `.detach()` calls, and the backward keeps $\gamma_i\Delta_i$ but drops the extremum terms of (2). LERP $K{=}32$ is the cleanest case because its forward is differentiable. Its matched run with full calibration gradients shows no detectable difference from softmax in this evaluation ($\Delta\mathrm{NLL} = -0.0008$, 95% CI $[-0.004, +0.002]$); the same forward with the detached backward ends $+0.737$ nats higher (Figure 2a). The failure is delayed: the run tracks softmax through 25M tokens and separates near 30M, where its row span also leaves the full-gradient run and then grows by more than an order of magnitude (Figure 2b). The contrast replicates on seeds 7 and 42 (Table S23), and Nearest fails at every K against its matched run (Figure 2c, Table S19).

Restoring zero-sum is not enough. Projecting each detached row onto the zero-sum subspace (P1) removes the common-mode component exactly, yet the projected run ends *worse* than plain detach (Figure 2a, Table S19).

Which extremum's gradient matters. Retaining only the maximum-dependent calibration gradient recovers almost the whole full–detach gap on both seeds; retaining only the minimum-dependent gradient recovers almost none of it (Figure 2d, Table S25). The maximum-only backward is not strictly zero-sum yet ends within 0.015 nats of full on both seeds, while the projected run above satisfies zero-sum and fails: zero-sum alone does not explain the outcomes, and a role for approximate

Table 2: The four formal suites and the question each answers. Each suite matches model, data and optimizer protocol (AdamW, cosine decay, 1024-token sequences, 131,072 tokens/step, no learning-rate or method tuning) and varies the operator (suite D also the seed); historical training-stack differences: Tables S2 and S4.

Suite	Purpose	Model	Tokens	Corpus	Seeds	Runs
A long-horizon	K, horizon, downstream	124M	2.5B	FineWebEdu-3B	1	15
B factorial	calibration $\times$ surrogate $\times K$	124M	250M	FineWebEdu-3B	1	10
C scale probe	ordering at $8\times$ parameters	1B	100M	WikiText-103	1	12
D seeds	5×5 grid, seed spread	124M	100M	WikiText-103	5	130

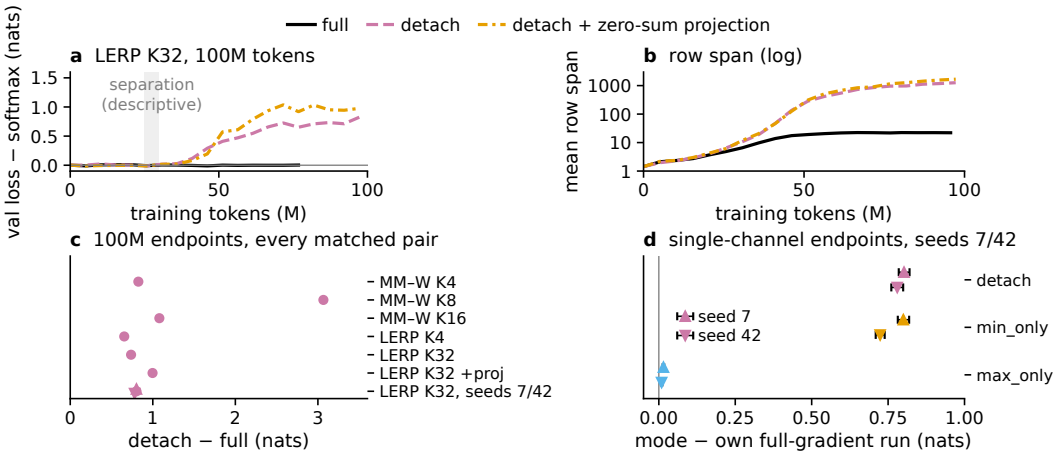


Figure 2: **Same forward, different calibration backward** (124M, 100M tokens, WikiText-103). (a) LERP $K{=}32$, seed 1337: logged validation loss minus softmax under full (black), detach (purple dashed) and detach with the zero-sum projection (orange dash-dot); grey band: observed separation. (b) Logged mean row span (log). (c) 100M endpoints, detach minus full, every matched pair (CIs: Tables S19, S23). (d) Single-channel ablation, LERP $K{=}32$: each run minus its own same-seed full run, seeds 7 ($\triangle$) and 42 ($\triangledown$), 95% block-bootstrap CIs over validation blocks (evaluation sampling only; Table S25).

Table 3: Matched $K{=}4$ factorial contrasts across the four suites (final checkpoint, nats; corpora as in Table 2); single-seed CI half-widths ≤ 0.0091, suite D paired by seed; interaction $= (FWM_P - FWM_W) - (\mathrm{MM}_P - \mathrm{MM}_W)$; intervals and condition-vs-softmax values in Table S7.

Contrast, $K{=}4$ (nats)	B: 124M@250M	A: 124M@2.5B	C: 1B@100M	D: 5 seeds, mean [95% t-CI]
Prob − Weight under MinMax, K4	-0.316	-0.829	-0.074	-0.071 [-0.087, -0.054]
Prob − Weight under FWM, K4	-0.032	-0.023	-0.022	-0.019 [-0.028, -0.009]
FWM − MinMax under Weight-STE, K4	-0.350	-0.848	-0.159	-0.176 [-0.193, -0.158]
FWM − MinMax under Prob-STE, K4	-0.067	-0.042	-0.108	-0.124 [-0.138, -0.109]
Interaction, K4	+0.284	+0.805	+0.051	+0.052 [0.030, 0.075]

zero-sum is not excluded. Fixed-upstream decompositions at 20M and 40M agree (Table S24). This identifies the dominant channel in this setting but does not separate the maximum's effects on score origin and grid width (Appendix F).

5.2 Calibration × surrogate interaction and seed robustness

Holding Nearest reconstruction and $K{=}4$ fixed, calibration and surrogate placement interact strongly (Table 3, Figure 3a). MinMax–Weight ends $+0.39$ nats above softmax at 250M tokens and $+0.89$ at 2.5B. Changing either factor, moving the surrogate to the probability level or replacing MinMax with FWM, removes most of the deficit; changing both adds little more, so the interaction is large and positive. The calibration effect, the surrogate effect and their interaction keep their sign at 1B parameters and on every seed of suite D.

MinMax→FWM changes the grid step and the tail treatment together; a single-seed control that keeps FWM's fixed step but extends the grid into the tail recovers most of the gap, so tail truncation is not required for the observed improvement in this setting; the control does not isolate resolution and is not cost-matched (Appendix D.4). A strict-forward replication at $K{=}4$ (seed 7; both surrogates retrained with a bit-exact hard forward) keeps the direction of both surrogate contrasts and of the interaction, and the size of the MinMax contrast (Table S6); one seed at one horizon on its own training stack, it does not extend to suites A–C (Appendix C.2).

The surrogate effect shrinks with K. In the 250M factorial the surrogate effect under FWM is about $30\times$ smaller at $K{=}16$ than at $K{=}4$ and only just detectable, whereas under MinMax it stays large (Figure 3b, Table S7); five seeds show the same decay (Figure 3d). MinMax–Weight $K{=}4$ is the only condition with an optimization abnormality (45% of steps clipped at 250M); MinMax–Weight $K{=}16$ shows none yet ends $+0.124$ nats behind softmax.

Seed robustness. Across five seeds the ordering of conditions is reproducible (Kendall's W 0.91; Figure 3c). Averaged over K, FWM reduces NLL by 0.067 nats relative to MinMax and Prob-STE

by 0.021 relative to Weight-STE, with a $+0.031$ interaction: the two changes substitute for each other. All three effects shrink with K, and the FWM cells at $K \geq 16$ lie within seed noise of softmax. In these historical runs the training stack was associated with the operator family (Appendix C).

5.3 K SCALING AND TRAINING HORIZON

LERP improves with training. LERP with the full gradient converges toward softmax over 2.5B tokens: K=4 goes from $+0.086$ at 125M to $+0.005$ at 2.5B (Figure 4b), and K=16 and 32 stay close to softmax from 250M on (final values: Figure 4a).

Reduced-precision exponential baselines. At 124M @ 100M (seed 7), BF16-exp ends -0.0034 nats from softmax and FP8-rounded exp (a weight-level straight-through surrogate with the fp32 exponential derivative, not surrogate-free) $+0.0028$, far below the same-seed LERP and MinMax K=4 gaps (Table S11). Like FWM, both apply a fixed rule in row-max-shifted coordinates that the learned span cannot widen, so precision alone does not explain the K-interval failures; they are not a controlled test of the hypothesis of §5.4 (Appendix D.5).

Larger K reduces the Nearest deficit, with diminishing returns. At 2.5B MinMax–Weight falls from $+0.89$ at K=4 to $+0.04$ at K=64 (Table S12); at 1B parameters (100M tokens) the final ΔNLL of LERP, MinMax–Weight and FWM–Weight orders the same way at every K; on five seeds the slope against $\log_2 K$ is negative for every family, with the high-K FWM cells within seed noise of softmax (Table S9).

The hard-vs-LERP ordering depends on the training budget. FWM–Weight K=4 minus LERP K=4 is negative at 125M and positive at 2.5B, FWM–Prob likewise (Figure 4c), and both are negative at 250M training tokens and at 1B parameters (100M tokens): LERP's gap keeps closing while the Nearest gaps plateau, so the relative NLL of LERP and Nearest changes with the training budget.

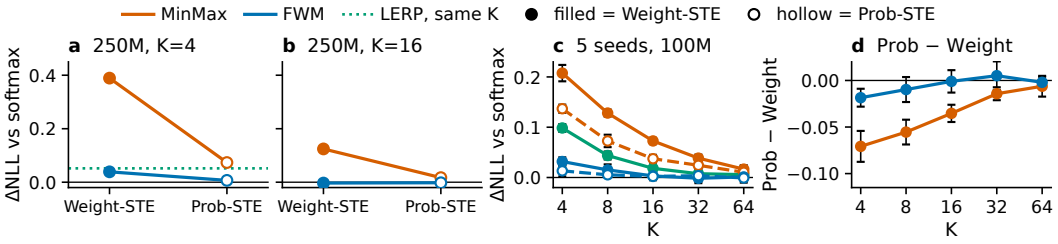


Figure 3: **Calibration $\times$ surrogate interaction for Nearest.** (a, b) 124M @ 250M at K=4 and K=16: final ΔNLL vs softmax by surrogate placement, one line per calibration, 95% block-bootstrap CIs (evaluation sampling only); dotted: LERP reference, in (a) only. (c) Five-seed mean ΔNLL vs K, all five families (suite D), 95% t intervals over seeds. (d) The surrogate effect Prob $-$ Weight against K under each calibration, paired by seed, 95% t intervals (4 df). Filled markers/solid lines: Weight-STE; hollow/dashed: Prob-STE.

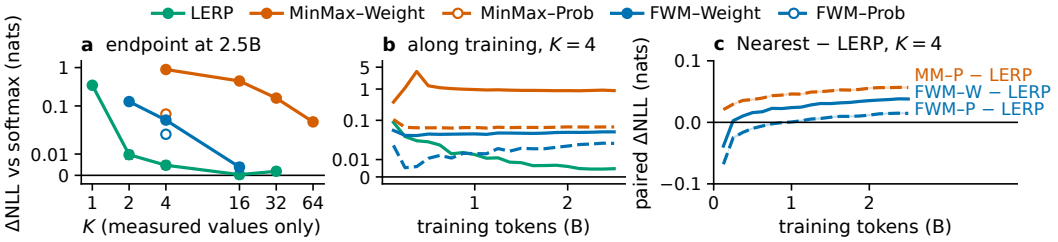


Figure 4: **124M @ 2.5B.** (a) Final ΔNLL vs softmax by K and family at the measured K only (whiskers: 95% block-bootstrap CI, evaluation sampling only). (b) Gap along training for the five K=4 conditions. (c) Nearest minus LERP at matched K=4; the 95% CI (evaluation sampling only) is narrower than the line width. (a, b): symlog vertical scale, linear within ± 0.02 nats; (c): linear.

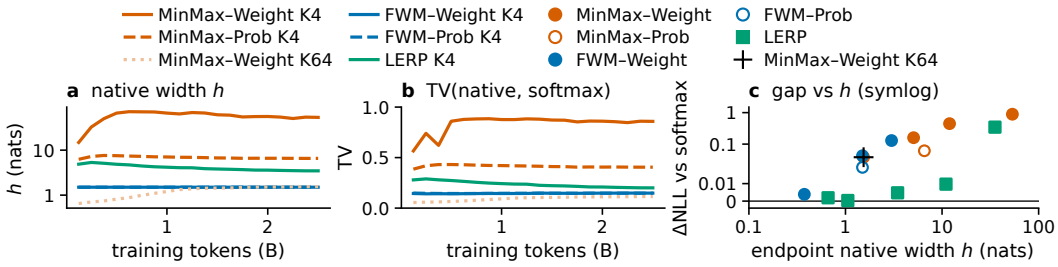


Figure 5: **Geometry and resolution at 2.5B.** Geometry statistics are layer means over eight validation blocks on each model's own scores. (a) Native interval width h. (b) Total variation between native and softmax probabilities. (c) Endpoint NLL gap versus native width for all 14 approximate conditions. The cross marks MinMax–Weight $K{=}64$ ($h = 1.54$, $\Delta\mathrm{NLL} = 0.038$): large score span, moderate interval width. Vertical axis symlog, linear for $|\Delta\mathrm{NLL}| \leq 0.02$.

5.4 Learned geometry: observations, counter-examples and a hypothesis

Among the formal conditions with full calibration gradients, MinMax–Weight is the only one whose score span keeps growing (Figure 5, Appendix D). At 2.5B its $K{=}4$ row span is about seven times softmax's, its native width reaches $h{=}53$ nats and the total variation between native and softmax probabilities is 0.86: the model sharpens scores that the operator cannot resolve. Moving the surrogate to the probability level with the same hard forward keeps the span at softmax's level (MinMax–Prob $K{=}4$), as does FWM, which changes calibration, tail and backward together; the span growth is specific to the surrogate–calibration combination.

Three observations limit the interpretation. MinMax–Weight $K{=}64$'s span more than doubles over training while its gap stays between $+0.02$ and $+0.04$ nats and h stays moderate (Figure 5c). At 250M the MinMax–Weight/Prob gap is already open at 50M tokens while the two geometries nearly coincide (Figure S5). MinMax–Weight $K{=}16$ has a gap without any gradient abnormality. Across the five-seed grid the condition-mean $\Delta\mathrm{NLL}$ has Spearman correlation $+0.95$ with native width and -0.31 with score span; among these designed conditions the width association is the stronger one (Figure S6).

The data fit a range–resolution feedback hypothesis: under MinMax–Weight a wider interval degrades the surrogate gradient, which lets the span grow further. No intervention in this paper identifies the mechanism; the associations above are descriptive.

5.5 Downstream transfer: graded, not binary

Predictions, metrics and three pre-specified groups (large-gap, intermediate-gap and near-baseline, by 2.5B $\Delta\mathrm{NLL}$) were fixed before any downstream run, and every condition is evaluated with its native forward (Appendix E). On WikiText-103, PTB and C4 the ordering is preserved (Spearman ≥ 0.95), and the near-baseline group shows small held-out differences of both signs: 10 of its 15 paired intervals exclude zero and 7 lie within ± 0.01 nats (Table S14). Substituting exact softmax at evaluation widens the gap in 37 of 42 cells; for LERP $K{=}2$ on the WikiText-103 test set it goes from $+0.005$ to $+1.814$ nats (Table S15): the trained weights depend on the training operator. The substitution was run on the corpora and probes, not on the benchmark battery.

On six benchmarks evaluated in seven configurations (Figure 6, Table S16), large corpus-NLL degradation transfers: all 35 large-gap comparisons (five conditions $\times$ seven configurations) have negative point estimates and 27 reach $q < 0.05$, and MinMax–Weight $K{=}4$ loses 3.9–19.4 accuracy points, less at larger K. A small NLL gap does not guarantee equality on every benchmark: of the near-baseline group's 35 comparisons, 8 reach $q < 0.05$, six negative (LERP $K{=}2/4/16/32$ on BLiMP; LERP $K{=}2$ on ARC-Easy and LAMBADA) and two positive (FWM–Weight $K{=}16$ on both SciQ configurations; Table S16); with one seed per condition these small differences cannot be attributed to the operator rather than to the run. FWM–Prob $K{=}4$ has the smaller training-domain gap than FWM–Weight $K{=}4$ but the larger copying gap at $L{=}128$ (single seed, synthetic probe; Table S18).

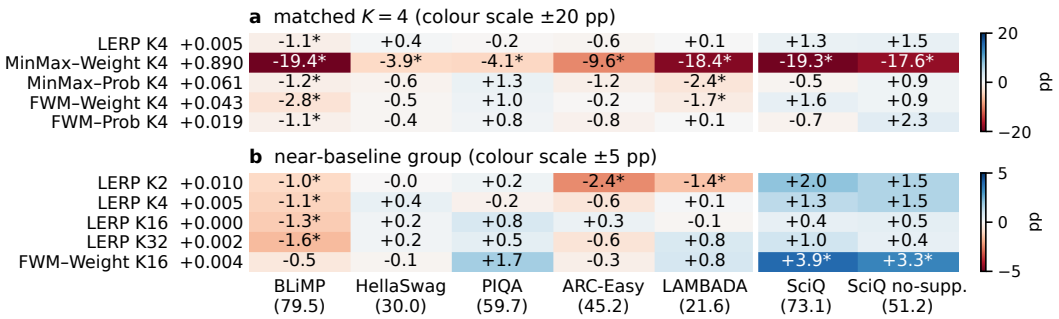


Figure 6: **Downstream accuracy differences from softmax** (pp, native forward), 2.5B endpoints; row labels: training-domain ΔNLL; column labels: softmax's accuracy. (a) The matched $K{=}4$ conditions ($\pm$20 pp scale); (b) the near-baseline group ($\pm$5 pp). BLiMP macro-accuracy; HellaSwag/PIQA/ARC-Easy `acc_norm`; LAMBADA and SciQ `acc`. Star: BH $q < 0.05$ within the column's 14-comparison family; all 14 conditions in Appendix E.

6 Discussion and Limitations

The same hard forward yields different models under different backward rules, and an incomplete Jacobian fails where the complete one does not. Whether the surrogate–forward mismatch can grow with the learned geometry is the hypothesis of §5.4: under MinMax h grows with the negative tail, whereas FWM fixes the width independently of it.

Limitations. *Seeds and budget.* Suites A–C use one seed; suite D's seed spread applies to its own setting, and its t intervals assume near-normal seed effects. The tail-policy comparison is single-seed (historical STE arithmetic, not equal cost), the channel ablation two seeds of one setting, and the strict-forward retraining and reduced-precision baselines one seed each at 100M tokens. No learning-rate sweep was run, so the 2.5B-token persistence is specific to the tested schedule.

Forward implementation and training stack. The two surrogates share the same forward mathematically; the historical implementations are not bitwise identical. Over the 12 audited frozen 124M $K{=}4$ endpoints, the largest absolute difference in mean validation NLL between a historical native forward and a direct hard forward is $3.4{\times}10^{-4}$ nats (Table S5): an evaluation-time quantity, not a per-block or training-time bound. Retraining with a bit-exact forward reproduces the direction of both contrasts and the size of the MinMax contrast at one seed and horizon (Appendix C.2), not repeated for suites A–C. In suite D the training stack was partly associated with the operator family, and the matched strict-forward check covers only seed 7 (Appendix C).

Window width, scale and horizon. $\tau = 6$ came from an inference-time screen on frozen weights whose final stage compared $\tau \in \{5, 6, 7\}$ on the WikiText-103 validation split later used for suites C and D; it is not shown optimal for training. The 1B suite is an early-regime probe on another corpus, with Prob-STE only at $K{=}4$, and no run exceeds 1B parameters or 2.5B tokens. Detectability and margin status remain separate labels (§4).

7 Conclusion

Pretraining with quantized softmax depends on both the forward approximation and its backward rule. Detaching the row extrema leaves the forward computation unchanged but causes delayed training failure; projecting the resulting gradients onto the zero-sum subspace does not recover the result with full calibration gradients. Under hard rounding at $K{=}4$, MinMax calibration combined with Weight-STE incurs a large loss gap. Replacing MinMax calibration with FWM or moving the surrogate to the probability level reduces this gap, with effects consistent in direction across the tested horizons, scales and seeds. Selected hard and interpolated configurations approach softmax validation loss. The remaining performance differences depend on the operator, K, training horizon and evaluation metric; small NLL gaps do not establish equivalence on downstream tasks. Approximate softmax operators for pretraining should therefore be evaluated through their calibration, reconstruction and backward rules together.

AI USE STATEMENT

Generative AI assistants (code-capable large language models) were used throughout this project, and we describe their role by activity. *Numerical results.* Every training run, evaluation, statistic, figure and table comes from the training and analysis code and the source data described in Appendix G; no number in the paper was generated by an AI model. *Reasoning and design.* AI assistants were used in discussions of research directions and of what the results support, in designing analyses (contrast definitions, bootstrap and pairing schemes, pre-declarations for the downstream evaluation), in drafting and revising analysis, evaluation, figure- and table-generation scripts and shell commands, in diagnosing experiments (including the gradient-consistency diagnostics that identified the detached-extremum defect), in choosing follow-up experiments, and in drafting, editing and restructuring the manuscript, its related-work survey and its responses to review. *Mathematical claims.* AI assistants were also used to derive and check the mathematical claims of §3 and Appendix B: the closed-form calibration derivatives, the zero-sum identity, the Weight-STE and Prob-STE backward rules and the derivative of the hard map. Every derivation was checked line by line by the authors, and each closed form is verified numerically against automatic differentiation of the frozen training code by the scripts named in Appendix B. *Author verification.* The authors set the research question, approved the experiments, executed or supervised the training and evaluation jobs, and reviewed the manuscript and its AI-assisted edits before submission. The degenerate-row behaviour is verified against the frozen code by a script that ships with the analysis code, and every number in the paper is regenerated from the stored result files by the figure and table scripts. Literature comparisons rest on the cited papers' stated methods and claims; we do not claim that every cited paper was read in full by every author. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies numerical attention operators on public text corpora (FineWebEdu, WikiText-103) and public benchmarks; it involves no human subjects, no personal data and no deployment. Its potential downstream use is more efficient training hardware, which carries the general dual-use considerations of any efficiency advance and no specific additional risk that we can identify.

REPRODUCIBILITY STATEMENT

Every run is identified by a frozen manifest carrying operator axes, seed, checkpoint paths and the SHA-256 of the training code root it is evaluated with (Appendix G); all checkpoints are evaluated through their own frozen code at batch size 1 (§4, Appendix C). The closed-form gradients of §3 are derived in Appendix B and verified against automatic differentiation by a script that ships with the analysis code. Figures and tables are generated from the per-checkpoint result files by two scripts; the τ selection record, the frozen-checkpoint forward discrepancy measurement and the detach intervention are documented in Appendices C and F. Code, manifests and per-block results will be made available to reviewers via an anonymous repository during the discussion period and publicly released upon acceptance.

REFERENCES

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: Parameterized clipping activation for quantized neural networks, 2018. arXiv:1805.06085.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023.

Siyu Ding, Mingchuan Ma, Jiabo Tong, Xingrun Xing, Ziming Wang, and Guoqi Li. Full-stack FP4: Stable LLM pretraining with quantized projections, optimizers, and attention, 2026. arXiv:2607.04422.

Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *International Conference on Learning Representations (ICLR)*, 2020.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, et al. A framework for few-shot language model evaluation, 2023. Zenodo, lm-evaluation-harness v0.4.

Alejandro Hernández-Cano, Dhia Garbaya, Imanol Schlag, and Martin Jaggi. Towards fully FP8 GEMM LLM training at scale, 2025. arXiv:2505.20524.

David T. Hoffmann, Simon Schrodi, Jelena Bratulić, Nadine Behrmann, Volker Fischer, and Thomas Brox. Eureka-moments in transformers: Multi-step tasks reveal softmax induced optimization problems. In *International Conference on Machine Learning (ICML)*, 2024.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Gamze Islamoglu, Moritz Scherer, Gianna Paulin, Tim Fischer, Victor J. B. Jung, Angelo Garofalo, and Luca Benini. ITA: An energy-efficient attention and softmax accelerator for quantized transformers. In *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 1–6, 2023. doi: 10.1109/ISLPED58423.2023.10244348. arXiv:2307.03493.

Sambhav R. Jain, Albert Gural, Michael Wu, and Chris H. Dick. Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks. In *Proceedings of Machine Learning and Systems (MLSys)*, 2020.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. I-BERT: Integer-only BERT quantization. In *International Conference on Machine Learning (ICML)*, 2021.

Shiwei Liu, Guanchen Tao, Yifei Zou, Derek Chow, Zichen Fan, Kauna Lei, Bangfei Pan, Dennis Sylvester, Gregory Kielian, and Mehdi Saligane. Consmax: Hardware-friendly alternative softmax with learnable parameters. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2024. arXiv:2402.10930.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

Asit Mishra, Dusan Stosic, Simon Layton, and Paulius Micikevicius. Recipes for pre-training LLMs with MXFP8, 2025. arXiv:2506.08027.

Houwen Peng, Kan Wu, Yixuan Wei, Guoshuai Zhao, Yuxiang Yang, Ze Liu, Yifan Xiong, Ziyue Yang, Bolin Ni, Jingcheng Hu, et al. FP8-LM: Training FP8 large language models, 2023. arXiv:2310.18313.

PyTorch contributors. torch.ao.quantization: Minmaxobserver and fakequantize, 2026. torch 2.12.0 source, `observer.py` and `fake_quantize.py`; observer statistics are computed on `x_orig.detach()`.

Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, and Russ Webb. Theory, analysis, and best practices for sigmoid self-attention, 2024. arXiv:2409.04431.

Kai Shen, Junliang Guo, Xu Tan, Siliang Tang, Rui Wang, and Jiang Bian. A study on ReLU and softmax in transformer, 2023. arXiv:2302.06461.

Moran Shkolnik, Maxim Fishman, Brian Chmiel, Hilla Ben-Yaacov, Ron Banner, and Kfir Yehuda Levy. EXAQ: Exponent aware quantization for LLMs acceleration. In *NeurIPS 2024 Workshop on Machine Learning and Compression*, 2024. arXiv:2410.03185.

Jacob R. Stevens, Rangharajan Venkatesan, Steve Dai, Brucek Khailany, and Anand Raghunathan. Softermax: Hardware/software co-design of an efficient softmax for transformers. In *Proceedings of the 58th ACM/IEEE Design Automation Conference (DAC)*, 2021. arXiv:2103.09301.

Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. Softmax is not enough (for sharp out-of-distribution), 2024. arXiv:2410.01104.

Mitchell Wortsman, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Replacing softmax with ReLU in vision transformers, 2023. arXiv:2309.08586.

Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D. Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-Dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities. In *International Conference on Learning Representations (ICLR)*, 2024.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

Zisheng Ye, Xiaoyu He, Maoyuan Song, Guoliang Qiu, Chao Liao, et al. BAPS: A fine-grained low-precision scheme for softmax in attention via block-aware precision rescaling, 2026. arXiv:2602.02071.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. In *International Conference on Learning Representations (ICLR)*, 2019.

Biao Zhang, Ivan Titov, and Rico Sennrich. Sparse attention with linear units. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Jintao Zhang, Jia Wei, Pengle Zhang, Jun Zhu, and Jianfei Chen. SageAttention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.02367.

Jintao Zhang, Marco Chen, Haoxu Wang, Kai Jiang, Ion Stoica, Joseph E. Gonzalez, Jianfei Chen, and Jun Zhu. SageBwd: A trainable low-bit attention, 2026. arXiv:2603.02170.

Yuan Zhang, Yonggang Zhang, Lele Peng, Lianghua Quan, Shubin Zheng, Zhonghai Lu, and Hui Chen. Base-2 softmax function: Suitability for training and efficient hardware implementation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 69(9):3605–3618, 2022.

Yuan Zhang, Lele Peng, Lianghua Quan, Yonggang Zhang, Shubin Zheng, and Hui Chen. High-precision method and architecture for base-2 softmax function in DNN training. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70(8):3268–3279, 2023.

Wanli Zhong, Haibo Feng, Zirui Zhou, Hanyang Peng, and Shiqi Yu. IntAttention: A fully integer attention pipeline for efficient edge inference. In *Proceedings of the 9th Conference on Machine Learning and Systems (MLSys)*, 2026. arXiv:2511.21513.

A OPERATOR DEFINITIONS AND EDGE CASES

All operators act on one causal row of fp32 scores $s = (s_1, \ldots, s_n)$, $S = QK^\top/\sqrt{d_h}$, computed inside a disabled-autocast region (fp32 tensors; TF32 tensor-core matrix products are enabled during training and disabled at evaluation); everything outside the attention kernel runs under bf16 autocast. Invalid (future) positions get $w = 0$. With $M = \max_j s_j$, $m = \min_j s_j$, $z_j = s_j - M$:

Softmax. $w_j^E = e^{z_j}$, $P^E = \mathrm{softmax}(s)$; explicit fp32 stable softmax.

MinMax grid. $z_{\min} = m - M$, $h = (M - m)/K$, $b_r = z_{\min} + rh$ for $r = 0, \ldots, K$; $r_j = \mathrm{clip}(\lfloor (z_j - z_{\min})/h \rfloor, 0, K-1)$, $t_j = \mathrm{clip}((z_j - b_{r_j})/h, 0, 1)$.

FWM (Fixed Window to Max; legacy identifier QRM, `range_mode=relevant_zero`). $h = \tau/K$, $b_r = -\tau + rh$, $\tilde{z}_j = \mathrm{clip}(z_j, -\tau, 0)$; $w_j = 0$ if $z_j < -\tau$; $\tau = 6$. The row minimum is computed only as a diagnostic.

LERP. $w_j^L = (1 - t_j)e^{b_{r_j}} + t_j e^{b_{r_j+1}}$: a continuous, interpolated weight, not one of the $K{+}1$ grid values. For $K{=}1$ a single interval interpolates between $e^{z_{\min}}$ and 1; the $K{=}1$ code root differs from the base root only by permitting $K{=}1$.

Nearest. $\ell_j = r_j + \mathbb{1}[t_j \geq \frac{1}{2}]$, $w_j^N = e^{b_{\ell_j}}$: the nearest grid *center* in z, not the nearest exponential value (nearest-center rounding). Under FWM the hard operator can additionally produce a zero tail.

Normalization. $P = w/\sum_l w_l$ after reconstruction, so $\sum_j P_j = 1$ in exact arithmetic for every operator.

Edge cases and conventions (as implemented). Read from the frozen `quant_attention.py` of every formal code root and of the legacy detach root, and checked numerically by `paper_training_v1/checks/check_degenerate_rows.py` (152 forward/backward cases, all finite, row sums 1 within 10^{-6}). Three conditions are kept apart: the nondegeneracy condition $h > 0$ under which the paper's formulas are stated; the implementation's regular branch ($M - m > \varepsilon$); and its degenerate branch ($M - m \leq \varepsilon$, $\varepsilon = 10^{-12}$). On the MinMax path (`_adaptive_interval_reconstruct`, identical in all roots) the extrema come from `amax`/`amin` over the causal-valid keys and a safe width $h_{\mathrm{safe}} = 1$ (degenerate) or h (regular) is formed before any division, so no $0/0$ is formed. Table S1 lists the conventions; all identities in the paper (secant slope, calibration derivatives, zero-sum) are stated for a nondegenerate row with $h > 0$, a locally fixed interval index and the tie convention of the table; no epsilon or span clamp other than ε exists.

Table S1: Edge cases: forward value and backward convention as implemented (verified finite in fp32 and fp64 for LERP, MinMax–Weight, MinMax–Prob at $K \in \{4, 16\}$ and for FWM–Weight/FWM–Prob).

Condition	Forward	Backward
Degenerate MinMax row ($M - m \leq \varepsilon$): single valid key or equal valid scores	weights set to 1 (`torch.where`), invalid keys 0, P uniform (softmax's forward)	zero gradient through `where`, extremum gradients cancel; finite
Very small positive span ($10^{-6} > \varepsilon$)	regular branch, tiny h, normal reconstruction	all gradients finite
FWM: single key or equal scores	$z_j = 0$ for every valid key, all at the top grid value, P uniform; no degenerate branch	regular formulas
Tied maxima/minima, positive span	unique-extremum formulas	`amax`/`amin` split the extremum gradient evenly among ties
Grid-boundary index switch	r_j is a floor, piecewise constant	zero derivative a.e.; LERP weight continuous, secant identities on either side
Nearest midpoint tie $t_j = \frac{1}{2}$	rounds up; hard weight discontinuous	(5) applies away from these points
FWM cutoff $z_j < -\tau$	masked to $w_j = 0$ after clamping; jump at $z = -\tau$	zero gradient (mask and clamp); statements apply to in-window keys

Unless explicitly labeled as a detached diagnostic (Appendix F), all formal runs include the full calibration gradient defined by their calibration rule; Table 1 therefore lists only the three design axes. FWM–LERP was not trained and appears only in the formula verification of Appendix B.

Under MinMax the interval width follows the learned row span, $h = \mathrm{span}/K$; under FWM it is the constant $h = \tau/K = 6/K$: at K=4, $h = 1.5$ nats under FWM against a quarter of the row span under MinMax (6–8 nats at softmax-like spans of 25–32, far more once the span expands).

B GRADIENT DERIVATIONS

Throughout, $g_j = \partial L/\partial P_j$ is the upstream gradient at the attention probabilities of one row, always evaluated at the forward (for Nearest: hard) probabilities. The value path $O = PV$ uses the forward P: $\partial L/\partial V = P^\top \partial L/\partial O$.

Normalization Jacobian. For $P = w/W$, $\partial P_i/\partial w_j = (\delta_{ij} - P_i)/W$, so $\gamma_j \equiv \partial L/\partial w_j = \frac{1}{W}\big(g_j - \sum_i P_i g_i\big)$.

Softmax. $\partial L/\partial s_j = P^E_j(g_j - \sum_i P^E_i g_i)$. The implementation subtracts M before exponentiating; autograd routes a gradient through M that cancels exactly by shift invariance.

LERP own-score slope. With the grid held fixed, $\Delta_j = \partial w^L_j/\partial z_j = (e^{b_{r_j+1}} - e^{b_{r_j}})/h$: the floor defining r_j is locally constant (zero derivative almost everywhere) and the clamp on t acts as the identity in the interior of an interval. Compared with the exact derivative e^{z_j}, Δ_j is constant within an interval.

Full calibration gradient (MinMax). Since $z_{\min} = m - M$ and $h = (M - m)/K$, the grid values b_{r_j}, b_{r_j+1} and the width h all depend on M and m. Writing $\rho_j = (r_j + t_j)/K \in [0, 1]$ for the normalized position of key j in $[z_{\min}, 0]$ and $X_j = \frac{1}{K}[(1 - t_j)r_j e^{b_{r_j}} + t_j(r_j{+}1)e^{b_{r_j+1}}]$, differentiating w^L_j through z_j, b_{r_j}, b_{r_j+1} and h gives (1). PyTorch's `amax`/`amin` split the extremum gradient evenly among ties. Adding the three per-key partials gives $\Delta_j + \partial w_j/\partial M + \partial w_j/\partial m = 0$ for every j, hence $\sum_i \partial L/\partial s_i = 0$.

Detach (legacy). $\partial L/\partial s_i|_{\mathrm{detach}} = \gamma_i \Delta_i$. Since $\sum_i \gamma_i \Delta_i \neq 0$ in general, each update carries a spurious common-mode component along a direction that cannot change the loss, and the relative gradients between keys are also wrong. The P1 control keeps the detached terms and projects each causal row onto the zero-sum subspace, $g'_{ij} = g_{ij} - \frac{1}{i+1}\sum_{k \le i} g_{ik}$, removing the common-mode component only.

Weight-STE. $w = w^L + \mathrm{sg}(w^N - w^L)$, $P = w/\sum_l w_l$. Forward $w = w^N$, $P = P^N$. Backward: the normalization Jacobian is evaluated at the hard point and the scalar derivatives (including the calibration terms) are those of LERP,

$$\frac{\partial L}{\partial s_i} = \gamma^W_i \Delta_i + \mathbb{1}[i{=}\arg\max] \sum_j \gamma^W_j \frac{\partial w^L_j}{\partial M} + \mathbb{1}[i{=}\arg\min] \sum_j \gamma^W_j \frac{\partial w^L_j}{\partial m}, \qquad \gamma^W_j = \frac{g_j - \sum_i P^N_i g_i}{W^N}.$$

Prob-STE. $P^N = \mathrm{sg}(w^N/W^N)$, $P^L = w^L/W^L$, $P = P^L + \mathrm{sg}(P^N - P^L)$. Forward $P = P^N$. Backward is the full Jacobian of normalized LERP at the LERP point: the same expression with $\gamma^P_j = (g_j - \sum_i P^L_i g_i)/W^L$.

FWM. w_j depends on the scores only through $\tilde{z}_j = \mathrm{clip}(s_j - M, -\tau, 0)$. For in-window keys $\partial w^L_j/\partial s_j = \Delta_j$ and $\partial w^L_j/\partial M = -\Delta_j$; tail keys have $w_j = 0$ and zero gradient through both the mask and the clamp; the row minimum plays no role. Thus $\partial L/\partial s_i = \gamma_i \Delta_i - \mathbb{1}[i{=}\arg\max] \sum_{j \in \mathrm{window}} \gamma_j \Delta_j$ with $\gamma = \gamma^W$ or γ^P (tail excluded from W^L, P^L), and the row gradient again sums to zero.

What differs between the surrogates. Both share the hard forward, g, the value-path gradient and Δ, $\partial w/\partial M$, $\partial w/\partial m$; they differ only in the point at which the normalization Jacobian is evaluated, $\gamma^W = (g - \langle g \rangle_{P^N})/W^N$ versus $\gamma^P = (g - \langle g \rangle_{P^L})/W^L$:

$$\gamma^W_j - \gamma^P_j = (g_j - \langle g \rangle_{P^N})\Big(\frac{1}{W^N} - \frac{1}{W^L}\Big) + \frac{\langle g \rangle_{P^L} - \langle g \rangle_{P^N}}{W^L}$$

vanishes when $w^N = w^L$ and otherwise depends on the within-interval positions of the scores (through $w^N - w^L$), on the reconstructed weights and probabilities and on g, so it is not proportional to h and need not be monotone in h. The main text uses only the qualitative statement that a wider interval permits a larger $|w^N - w^L|$; the empirical decay of the surrogate effect with K is a measured result, not a consequence of these formulas. Under FWM $h = \tau/K$ regardless of the learned tail, and tail keys contribute to neither W.

Derivative of the full hard map. Under MinMax the grid value at level ℓ is $b_\ell = (1-\ell/K)(m-M)$, so with the index ℓ_j held fixed (away from index switches and midpoint ties) and unique extrema, $\partial b_{\ell_j}/\partial m = c_j$ and $\partial b_{\ell_j}/\partial M = -c_j$ with $c_j = 1 - \ell_j/K$, and

$$\partial w_j^N/\partial s_i = c_j\, w_j^N \big(\mathbb{1}[i{=}\arg\min s] - \mathbb{1}[i{=}\arg\max s]\big), \tag{5}$$

generally nonzero (a two-key row is exactly the two-element softmax). This derivative is *not* what any trained operator back-propagates: in both STE implementations the hard branch enters only inside the stop-gradient (as $w^N - w^L$ or $P^N - P^L$), so its dependence on M and m is cut and the calibration derivatives that reach s are those of w^L. Under FWM $b_\ell = -\tau + \ell h$ is data-independent and the hard weights are locally constant when index and tail mask are fixed.

Verification. `paper_training_v1/checks/check_minmax_hard_jacobian.py` checks (5) in float64 on 100 random rows away from boundaries (max deviation $2.8{\times}10^{-17}$ from autograd, $8.8{\times}10^{-11}$ from central finite differences), the two-key softmax identity (10^{-16}) and the zero Jacobian of the FWM hard map; `report/verify_gradient_formulas.py` compares the closed forms above with autograd of the frozen `quant_attention.py` on 200 random float64 rows for MinMax and FWM $\times$ {LERP, Weight-STE, Prob-STE}: all agree to $< 10^{-9}$, and every autograd row gradient sums to zero. The diagnostics run when the detach defect was found are listed in Appendix F.

C Protocol details

C.1 Training, hardware and conditions

Training. Within a suite every run shares the model, data and optimizer protocol, verified from every final checkpoint's config and re-asserted by the evaluator on every checkpoint; execution-stack and switch differences are recorded in Tables S2 and S4 and below. 124M: 12 layers, 12 heads, d=768, 1024-token context. 1B: 32 layers, 24 heads, d=1536 (≈983M parameters). AdamW, β_1=0.9, β_2=0.95, weight decay 0.1, gradient clip 1.0, linear warmup then cosine decay to $0.1\times$ peak, micro-batch 1 with gradient accumulation to 131,072 tokens per step. Peak LR $6{\times}10^{-4}$ (124M) and $3{\times}10^{-4}$ (1B). Warmup 25M tokens (250M and 2.5B suites), 10M (100M suites). Final steps: 1908 (250,085,376 tokens), 19,074 (2,500,067,328), 763 (100,007,936). Two 2.5B runs (MinMax–Prob K=4, FWM–Prob K=4) were resumed after host outages at 500.0M and 625.1M tokens; saved checkpoints are unaffected and the trajectories show no discontinuity.

Numerics and hardware. Every training script enables TF32 (verified in the frozen code of every training root), so $QK^\top$ and PV are TF32 tensor-core products during training while the operator itself (calibration, reconstruction, normalization, surrogate) runs on fp32 tensors inside a disabled-autocast region; everything outside attention runs under bf16 autocast; evaluation and all diagnostics use the same split with TF32 off. Table S4 lists training and evaluation hardware per suite as far as the run records state it: environment records exist for the 4×RTX 5090 pods and the local RTX 3090; some cloud pods logged only the host name, and their GPU model is author-confirmed as RTX 5090 (rented pods; the confirmation is the authors' record, not a pod log); software versions that were not logged remain "not recorded". GPU hours come from the logged throughput of each run (training only, approximate). Gradient checkpointing: six suite-D runs on seed 1337 (softmax, LERP K=4/8/16/32, MinMax–Weight K=32) were trained with it enabled and the other 124 without (checkpoint configs, `manifests/local_training_asset_audit_20260925.tsv`); step-0 gradient equivalence with and without it was checked on the inputs of Appendix F only. Table S2 splits suite D by operator family and training stack: all 50 Prob-STE runs and the FWM–Weight K=32/64 runs trained on cloud RTX 5090 pods, the other families on the local RTX 3090, so training stack is associated with operator family in suite D and the historical surrogate contrasts

also reflect training-stack differences; same-GPU evaluation controls evaluation-stack differences only. The strict-forward E2 runs (Appendix C.2) give a matched comparison within one training stack at seed 7; their differences from the historical runs (Table S6) combine training stack and forward implementation and do not isolate a hardware effect. The reduced-precision baselines took 2 h 14 min and 2 h 18 min on the local RTX 3090 (observed durations).

Table S2: Suite D: operator family by training stack, in complete training runs (130 formal runs; supplementary runs excluded and listed in Table S4). "Logged": GPU model recorded by the pod's `nvidia-smi` in the master log; "author-confirmed": rented RTX 5090 pods whose logs did not record the model. Local runs: RTX 3090, torch 2.5.1+cu121.

Family	local	pod, logged	pod, confirmed	runs
softmax	5	0	0	5
LERP	25	0	0	25
MinMax–Weight	25	0	0	25
MinMax–Prob	0	5	20	25
FWM–Weight	15	0	10	25
FWM–Prob	0	5	20	25
all	70	10	50	130

Table S3: Secondary frozen-checkpoint audit (historical expressions only): Prob-STE frozen code minus the shared Weight-STE-expression evaluator, Δ per block on the full validation split (976 blocks for suites A and B, 243 for C); mean Δ over blocks and the largest single-block $|\Delta|$. Neither side is a direct hard forward (Table S5); this audit alone covers suite C and $K{=}16$. Absolute NLL, mean $|\Delta|$ and the first-8-block maximum of the parity check: file `tab_rounding_full_split_audit.tex` in `compact/migrated/`.

Run (suite: condition)	mean Δ	max $\lvert\Delta\rvert$
B: MinMax–Prob K4	$3.28\text{e}{-5}$	$2.14\text{e}{-2}$
B: MinMax–Prob K16	$-3.37\text{e}{-5}$	$8.90\text{e}{-3}$
B: FWM–Prob K4	$1.08\text{e}{-4}$	$9.90\text{e}{-3}$
B: FWM–Prob K16	0	0
A: MinMax–Prob K4	$-9.91\text{e}{-5}$	$1.38\text{e}{-2}$
A: FWM–Prob K4	$2.33\text{e}{-5}$	$1.41\text{e}{-2}$
C: MinMax–Prob K4	$3.48\text{e}{-4}$	$1.36\text{e}{-2}$
C: FWM–Prob K4	$-3.58\text{e}{-4}$	$7.98\text{e}{-3}$

Table S4: Training and evaluation hardware per suite and supplementary group, from the run manifests and environment records. Stack IDs: S1 = RTX 5090, torch 2.13.0+cu132, py3.12 (environment record); S2 = rented RTX 5090 pod whose log did not record the GPU model (author-confirmed), torch not recorded; S3 = local RTX 3090, torch 2.5.1+cu121, py3.11; S4 = RTX 5090 per pod log, torch not recorded. Evaluation on host7 (RTX 5090) or the local RTX 3090. GPU-h: approximate single-GPU wall-clock hours from the training logs (training only; the reduced-precision row is the observed duration of its two runs).

Suite / group	Runs	Training stack (one GPU per run)	Evaluation GPU	GPU-h
A (124M @ 2.5B)	15	S1 (13 runs); S2 (2 Prob-STE runs)	RTX 5090	763
B (124M @ 250M)	10	S1 (5 runs); S2 (5 runs)	RTX 5090	56
C (1B @ 100M)	12	S1 (10 runs); S2 (2 WikiText Prob-STE reruns)	RTX 5090	245
D (124M @ 100M, 5 seeds)	130	S3 (70 runs); S4 (10 runs); S2 (50 runs); family split: Table S2	RTX 3090	498
Historical detach record (§5.1)	16	S3	RTX 3090	—
Detach replication, seeds 7/42 (App. F)	2	S3	RTX 3090	11
Single-channel E1, seeds 7/42 (App. F)	4	S3	RTX 3090	18
Strict-forward E2 (App. C.2)	4	S1	RTX 5090 and RTX 3090	11
Tail-policy (App. D.4)	2	S3	RTX 3090	16
Reduced-precision exponential baselines (App. D.5)	2	S3	RTX 3090	4.5

Conditions per suite. A (124M @ 2.5B, 15 runs): softmax; LERP $K{=}1/2/4/16/32$; MinMax–Weight $K{=}4/16/32/64$; MinMax–Prob $K{=}4$; FWM–Weight $K{=}2/4/16$; FWM–Prob $K{=}4$; 20 checkpoints per run. B (124M @ 250M, 10 runs): softmax; LERP $K{=}4$; {MinMax, FWM}$\times${Weight-STE, Prob-STE}$\times\{K{=}4, 16\}$; 10 checkpoints per run. C (1B @ 100M, 12 runs): softmax; {LERP, MinMax–Weight, FWM–Weight}$\times\{K{=}4, 16, 32\}$; MinMax–Prob $K{=}4$; FWM–Prob $K{=}4$; 5 checkpoints per run. D (124M @ 100M, 130 runs): softmax and {LERP, MinMax–Weight, MinMax–Prob, FWM–Weight, FWM–Prob}$\times\{K{=}4, 8, 16, 32, 64\}$ on seeds 7/42/253/1337/2026; 5 checkpoints per run. Formal total 167 runs and 1,106 evaluated checkpoints (Appendix G); reproduction tolerances of the endpoint evaluations are given under "Common limits" in Appendix C.2.

Corpora. FineWebEdu-3B and WikiText-103, GPT-2 BPE. Validation: 976 non-overlapping 1024-token blocks (FineWebEdu) and 243 (WikiText-103); a block is $x = \mathrm{data}[kT{:}kT{+}T]$, $y = \mathrm{data}[kT{+}1{:}kT{+}T{+}1]$.

C.2 FORWARD IMPLEMENTATION: THREE COMPARISONS KEPT APART

Identity statements. (i) *Mathematical*: Weight-STE and Prob-STE define the same hard forward P^N. (ii) *Implementation*: a single shared operator implementation, which evaluates the Weight-STE expression $w^L + (w^N - w^L)$, reproduces the frozen softmax, LERP $K{=}1$ and Weight-STE code bit for bit on the parity blocks, but not the Prob-STE code, which evaluates $P^L + (P^H - P^L)$ in fp32 with P^H itself normalized from the Weight-STE expression. Neither historical surrogate evaluates P^N bit-exactly on GPU at trained score scales, and Nearest models amplify such differences through rounding decisions in later layers, so the Prob-STE runs were trained on a forward that differs from the Weight-STE forward at the rounding level: a confound of the surrogate contrast that frozen-code evaluation reports but does not remove. The detach/full modes instead share one forward code path differing only by `.detach()` on the row extrema, so their forwards are identical by construction. Batch 1 vs batch 8 differs by up to $8.1{\times}10^{-3}$ on single blocks, which is why every number uses batch 1. The three comparisons below concern floating-point implementations of one mathematical forward and change no formal number; each states its object, scope, result and own limit, and the shared limits follow once.

(i) Historical Prob-STE code vs the shared Weight-STE expression (secondary audit). *Object*: the two historical expressions against each other, not against a direct hard forward. *Scope*: all 25 Nearest runs of suites A–C, evaluated twice on their full validation split (976 or 243 blocks, batch 1, same numerics) with the frozen training code and with the shared implementation; the only comparison covering suite C and $K{=}16$. *Result*: the 17 Weight-STE runs read exactly zero on every block, so the shared implementation matches the Weight-STE code; for the 8 Prob-STE runs (Table S3) the difference in mean NLL is at most $1.1{\times}10^{-4}$ nats on the 976-block FineWebEdu splits and $3.6{\times}10^{-4}$ on the 243-block WikiText split, with inconsistent sign (four positive, three negative, one exactly zero) and per-block differences that reach $2.1{\times}10^{-2}$ but cancel in the mean: a frozen-checkpoint forward discrepancy, much smaller than the MinMax $K{=}4$ surrogate contrasts (-0.32 to -0.83 nats) and the pooled five-seed surrogate effect (-0.021), comparable to the smallest contrasts discussed (about 0.001, FWM at $K{=}16$). *Limit*: it does not bound the deviation of either surrogate from P^N, and the single-block maximum is an observed value, not a sensitivity bound.

(ii) Historical native forward vs a direct hard forward on frozen endpoints (supplement). *Object*: each historical implementation against a direct hard forward on the same trained weights. In the historical code the hard value is never computed directly (Weight-STE evaluates $w^H = w^L + \mathrm{sg}(w^N - w^L)$ and $P^H = \mathrm{normalize}(w^H)$; Prob-STE evaluates $P = P^L + \mathrm{sg}(P^H - P^L)$ with that same nested P^H; $P^H = P^N$ in exact arithmetic but not bitwise), so a strict implementation must replace both the weight-level and the probability-level expression. *Scope*: 12 frozen $K{=}4$ endpoints (suites A, B and D seed 7; MinMax and FWM; Weight-STE and Prob-STE; 124M), each evaluated twice on one RTX 3090 with its own frozen code root, native and with only the surrogate combination rewritten to $\mathrm{sg}(x^N) + (x^L - \mathrm{sg}(x^L))$ at both levels, whose forward was verified bit-identical to the directly computed $w^N / \sum w^N$ on random rows and on the real scores of every layer (12 of 12). *Result* (Table S5): replacing the historical forward by the direct hard forward changes mean validation NLL by at most $3.4{\times}10^{-4}$ nats (largest observed $3.32{\times}10^{-4}$), while single blocks differ by up to 0.026 nats, so the mean bound is not a per-block bound. *Limit*: an evaluation-time effect at these frozen checkpoints on this stack, not extending to suite C, $K{=}16$ or unevaluated checkpoints; the two CIs that exclude zero are not a finding, nor the ten that include zero bitwise equivalence. The formal tables keep the native evaluations.

(iii) Retraining with a strict hard forward (supplement). *Object*: whether the surrogate contrast depends on the rounding of the historical Prob-STE forward. *Scope*: the four $K{=}4$ cells of suite D retrained at seed 7 (124M, WikiText-103, 100M tokens) with a forward that returns P^N bit for bit, $P = \mathrm{sg}(P^N) + \big(P^L - \mathrm{sg}(P^L)\big)$ for Prob-STE and $w = \mathrm{sg}(w^N) + \big(w^L - \mathrm{sg}(w^L)\big)$, $P = w / \sum_l w_l$, for Weight-STE, from one shared hard reconstruction (P^N detached because the MinMax hard branch still carries a gradient through the extrema; parentheses fix the evaluation order); the identity was verified against the direct P^N on synthetic, boundary and real seed-7 scores on CPU and GPU, and the two placements give bit-identical forwards at equal parameters. Both surrogates were retrained because the historical Weight-STE forward is itself not bit-identical to w^N on GPU at trained score scales (the subtraction leaves the Sterbenz range as h grows under MinMax; under FWM a 1-ulp

Table S5: Frozen-endpoint audit: historical native forward minus direct hard forward, 12 $K{=}4$ endpoints (124M; suites A and B on their 976-block FineWebEdu split, suite D seed 7 on the 243-block WikiText-103 split), both forwards on one RTX 3090 with the endpoint's own frozen code root. Δ mean NLL in nats with 95% circular block-bootstrap CI (block 16, 4000 resamples); last column: largest single-block $|\Delta|$. Largest $|\Delta|$ mean over the 12 endpoints: $3.32{\times}10^{-4}$. Data: `weight_native_vs_strict.json` and `prob_native_vs_strict.json` in `round2/task1` of `supplement_four_runs_20260923`.

Surrogate	Suite	Calib.	blocks	Δ mean NLL	95% CI	max $\lvert\Delta\rvert$ per block
Weight-STE	A: 124M @ 2.5B	FWM	976	3.04e−5	[−6.36e−5, 1.25e−4]	8.42e−3
Weight-STE	A: 124M @ 2.5B	MinMax	976	−3.32e−4	[−6.40e−4, −2.46e−5]	2.61e−2
Weight-STE	B: 124M @ 250M	FWM	976	1.00e−4	[5.60e−6, 1.93e−4]	8.68e−3
Weight-STE	B: 124M @ 250M	MinMax	976	−8.29e−5	[−4.56e−4, 2.91e−4]	2.26e−2
Weight-STE	D: 124M @ 100M, seed 7	FWM	243	1.03e−5	[−1.38e−4, 1.68e−4]	6.47e−3
Weight-STE	D: 124M @ 100M, seed 7	MinMax	243	−2.12e−4	[−6.54e−4, 2.53e−4]	8.70e−3
Prob-STE	A: 124M @ 2.5B	FWM	976	−8.41e−5	[−2.69e−4, 1.02e−4]	1.12e−2
Prob-STE	A: 124M @ 2.5B	MinMax	976	−4.58e−5	[−3.00e−4, 2.17e−4]	1.43e−2
Prob-STE	B: 124M @ 250M	FWM	976	−1.34e−4	[−3.38e−4, 5.73e−5]	1.13e−2
Prob-STE	B: 124M @ 250M	MinMax	976	−4.13e−5	[−3.83e−4, 2.91e−4]	2.39e−2
Prob-STE	D: 124M @ 100M, seed 7	FWM	243	−1.09e−4	[−3.78e−4, 1.69e−4]	8.83e−3
Prob-STE	D: 124M @ 100M, seed 7	MinMax	243	2.46e−4	[−1.53e−4, 6.24e−4]	1.09e−2

CPU/GPU difference in $e^{-4.5}$). Training ran on an RTX 5090 with torch 2.13, unlike the RTX 3090 / torch 2.5.1 of the seed-7 controls; endpoints were evaluated on both GPUs. *Result* (Table S6): the direction of both contrasts and the MinMax magnitude are unchanged; the larger FWM value comes mostly from the new Prob-STE run (-0.012 against the historical one). *Limit*: one pair of runs cannot separate rounding, environment and training nondeterminism (identical-code replays on the same hardware diverge within 50 steps), so the change is not attributed to the forward; the historical five-seed SD for that cell (0.013) is background only; the small single-seed advantage of the new FWM–Prob run over softmax is not a general improvement; the runs are supplementary, never merged into the five-seed tables, and the training-time effect in suites A–C remains unmeasured.

Table S6: Strict-forward replication at $K{=}4$ (hard Nearest, 124M, WikiText-103, 100M tokens, seed 7; ΔNLL in nats, paired per validation block, 95% circular block-bootstrap CIs over 243 blocks, evaluation sampling only). Interaction follows Table 3, $(FWM_P - FWM_W) - (\mathrm{MM}_P - \mathrm{MM}_W)$. Groups: strict pairs on the training GPU; the same pairs on a secondary evaluator; comparisons against the historical seed-7 runs on frozen weights; historical background, which is not a same-stack control.

Contrast (nats)	Evaluator	Δ	95% CI
Strict pairs: the four E2 runs, strict-hard forward, evaluated on the training GPU			
MinMax: Prob − Weight (strict pair)	RTX 5090	-0.0770	[-0.0812, -0.0729]
FWM: Prob − Weight (strict pair)	RTX 5090	-0.0390	[-0.0422, -0.0359]
Interaction $(\mathrm{FWM}_P - \mathrm{FWM}_W) - (\mathrm{MM}_P - \mathrm{MM}_W)$	RTX 5090	+0.0381	[+0.0331, +0.0425]
Secondary evaluator: the same strict pairs re-evaluated on the RTX 3090			
MinMax: Prob − Weight (strict pair)	RTX 3090	-0.0772	[-0.0816, -0.0731]
FWM: Prob − Weight (strict pair)	RTX 3090	-0.0388	[-0.0423, -0.0356]
Interaction, as above	RTX 3090	+0.0384	[+0.0331, +0.0433]
Historical comparisons: new strict runs vs the historical seed-7 runs (frozen weights, different training stack)			
MinMax: new Prob − historical Prob, strict forward on both	RTX 5090	+0.0008	[-0.0024, +0.0040]
FWM: new Prob − historical Prob, strict forward on both	RTX 5090	-0.0121	[-0.0154, -0.0090]
MinMax: new strict Weight − historical Weight	RTX 3090	-0.0017	[-0.0047, +0.0012]
FWM: new strict Weight − historical Weight	RTX 3090	+0.0030	[+0.0001, +0.0059]
MinMax: historical Prob, native − strict forward (frozen weights)	RTX 5090	-0.0001	[-0.0005, +0.0003]
FWM: historical Prob, native − strict forward (frozen weights)	RTX 5090	+0.0000	[-0.0003, +0.0003]
Background: historical seed-7 runs only (not a same-stack control)			
MinMax: historical Prob − historical Weight, seed 7	RTX 3090	-0.0793	[-0.0834, -0.0752]
FWM: historical Prob − historical Weight, seed 7	RTX 3090	-0.0240	[-0.0264, -0.0216]

Common limits. Neither (i) nor (ii) bounds accumulated training-time effects, and (iii), the only training-time evidence, covers neither suites A–C nor other seeds. Every CI above covers evaluation sampling only. Reproduction tolerances refer to specific checks: re-evaluating each suite's endpoints against its earlier independent evaluation agreed within $4.5{\times}10^{-5}$ nats except MinMax–Weight $K{=}4$ ($5.4{\times}10^{-4}$ at 2.5B, $3.9{\times}10^{-4}$ at 1B). Across stacks they do not hold: in the measured RTX 3090 / RTX 5090 comparisons (different GPUs and torch versions) mean NLL of the same checkpoint differed by up to about $5{\times}10^{-4}$ nats with a maximum per-block difference of 0.0125 (`supplement_four_runs_20260923/reports/supplement_e1_recovery_and`

_eval_shift.json), an observed range, not a bound. Same-GPU pairing keeps such offsets out of a contrast but does not make the native−strict difference itself invariant across GPUs (for suite D MinMax–Prob its sign differs between the two GPUs); contrasts of about 0.001 nats are therefore read only within one evaluation stack.

C.3 Selection of τ, statistics and evaluation

Selection of τ (multi-stage). $\tau = 6$ was selected before any FWM training run by an inference-time screen on frozen softmax-trained weights: the suite-A softmax endpoint (formal_124m_s1337_exact_2p5b/ckpt_step019074_2500.1m.pt), operator swapped at inference, bf16 forward; record in analysis_main_v1/provenance/tau_selection/. The split was the WikiText-103 validation file (sha256 397ae25d...) on which suites C and D are later evaluated, so WikiText-103 validation participated in the selection; suites A and B are evaluated on a different corpus. Stages: (1) floor tail ($z \leftarrow -\tau$), 32 blocks, $K \in \{4, 16, 32\}$, $\tau \in \{8, 12, 16, 20, 24, 32\}$, τ=8 best; (2a) floor tail, $\tau \in \{4, 5, 6, 7, 8, 10, 12\}$, τ=8 best and collapse below 8; (2b) zero tail (w=0), same candidates, τ=7 at K=4 and τ=6 at K=16, 32; (3) zero tail, all 243 blocks, $K \in \{4, 8, 16, 32\}$, $\tau \in \{5, 6, 7\}$. Stage-3 NLL (softmax 4.3119; 10,000-resample block bootstrap CIs recorded per condition):

K	MinMax	τ=5	τ=6	τ=7	τ=6 − softmax [95% CI]
4	5.8590	4.3655	**4.3613**	4.3670	$+0.0494$ [$+0.0467$, $+0.0522$]
8	5.3821	4.3311	**4.3244**	4.3285	$+0.0125$ [$+0.0113$, $+0.0139$]
16	4.7333	4.3224	**4.3132**	4.3150	$+0.0013$ [$+0.0007$, $+0.0020$]
32	4.3765	4.3200	**4.3111**	4.3115	-0.0008 [-0.0014, -0.0001]

The procedure is reproducible and inference-time only: it does not establish that $\tau = 6$ is optimal for training, and the wider candidates were eliminated on a 32-block subset. Two inference-time studies on other models report windows of the same scale (IntAttention (Zhong et al., 2026): $c = 6.6$, stable for $c \in [5.5, 7.7]$; EXAQ (Shkolnik et al., 2024): optimal clipping 3.3–8.0 nats); neither is a training result.

Contrasts and intervals. A contrast $\sum_c a_c \, \mathrm{NLL}_c$ is computed per block and averaged; 2×2 main effects average over the other factor with weights $\pm\frac{1}{2}$ and the interaction is $(FWM_P - FWM_W) - (\mathrm{MM}_P - \mathrm{MM}_W)$. Single-seed CIs: circular moving-block bootstrap, block length 16, 4000 resamples, 95% percentile. Five-seed: paired differences by seed, 95% t interval (4 df), exact enumeration of all 2^5 sign assignments for the permutation p (floor $2/32$), BH q across the 25 conditions.

Geometry probe. Forward pre-hooks on each attention layer recompute the fp32 scores exactly as the operator does on the first 8 validation blocks (rows $i \geq 1$; averaged over rows, reported per layer and as a layer mean): span $M - m$; native width h; 0.1% quantile of z; per-token query/key norms; softmax entropy on the model's own scores; fraction of entries with $z < -6$; top-bucket mass (softmax mass on keys in the operator's top level or top interval); effective buckets $\exp H(q)$ with $q_b \propto \sum_{j \in b} P_j^E$; $\mathrm{TV} = \frac{1}{2} \sum_j |P_j^{\mathrm{native}} - P_j^E|$.

Five-seed identity. The surrogate is not recorded in the checkpoint config (a Weight-STE and a Prob-STE run of the same calibration have identical config fields); identity is established by the SHA-256 of quant_attention.py in the training code root, matched byte for byte to the evaluation snapshots (Appendix G). The integrity gate checked 130 runs and 646 evaluation jobs with 0 errors (the legacy seed-1337 softmax run has only its endpoint checkpoint, hence 646 rather than 650); its frozen exclusion list and the dated addendum are described in Appendix G.

Downstream (2.5B suite only). Pre-declared condition groups, predictions, a ± 0.01-nat practical-equivalence margin for NLL-type metrics only, and paired-bootstrap decision rules were fixed before any downstream number was produced. Each NLL-type comparison carries two independent labels, *detectability* (the paired 95% CI excludes zero / includes zero) and *margin status* (the CI lies wholly inside the pre-declared ± 0.01-nat margin / does not); "within the pre-declared margin" is used only for the first case of the second label, "no detectable difference" only when the CI includes zero, never as a synonym for equivalence; no margin is declared for accuracy metrics, so accuracy results are

never called equivalent; split-half fluctuation is a descriptive scale, not a threshold. Layer-1 q values are BH within the 14 condition-vs-softmax comparisons of each corpus and forward mode. *Layer 1*: WikiText-103 test (276 blocks), PTB test (102), C4 validation (302 documents, document-cluster bootstrap); split-half fluctuation 0.057/0.072/0.131 nats. *Layer 2*: three synthetic probes from random token sequences with a fixed generator seed: A, induction (a bigram (A, B) inserted, A recurs at distance $d \in \{64, \ldots, 896\}$, predict B; 400 families $\times$ 5 distances = 2,000 induction samples, each scored with a matched control sequence without the earlier occurrence, 4,000 scored sequences in all); B, key–value retrieval (16–64 pairs, near/far queries, 2,400 samples); C, prefix copying (a random segment S of length $L \in \{32, 64, 128\}$, a separator, then S again, the second copy scored per token; 300 samples per L, 900 in total); scores are per-token target NLL and top-1 accuracy, and only probe C had adequate resolution at 124M. *Layer 3*: lm-evaluation-harness 0.4.13 (Gao et al., 2023), zero-shot, batch size 1, bf16 autocast outside the fp32 attention kernel, fp32 logits; BLiMP (task `blimp`, validation, a fixed 200-pair subset of each of the 67 paradigms selected by the harness `limit` argument with seed 20260920, 13,400 pairs), HellaSwag (validation, 10,042), PIQA (validation, 1,838), ARC-Easy (test, 2,376), LAMBADA (`lambada_openai`, test, 5,153), SciQ (test, 1,000) with the support passage and a no-support variant (task YAML with the passage removed); comparisons paired per item on identical items in identical order (paired bootstrap on per-item differences, 4000 resamples; permutation p, 20,000; BH within each configuration $\times$ metric family of 14 comparisons); nothing is combined into a single score. Task selection proceeded in three recorded stages on 2026-09-20: stage A scored the softmax 2.5B endpoint alone on nine benchmarks in ten configurations (the seven above plus WinoGrande, ARC-Challenge and OpenBookQA); the pre-registered stage-B screen excluded those three because softmax scored below chance on them (4.7–5.2 SE), after seeing softmax-only results and before any approximate condition was scored; the stage-C battery of seven configurations was fixed in its manifest before the first condition ran, and no task or metric was added, dropped or re-weighted afterwards (`downstream_v1/REPORT.md`, §1.7, §6, §7).

D SEED ROBUSTNESS, HORIZON AND SCORE GEOMETRY: EXTENDED RESULTS

D.1 FULL CROSS-SUITE CONTRAST TABLE

Table S7: Every matched contrast of Table 3 for all four suites with its interval, plus the main effects, the $K{=}16$ terms of the 250M factorial and the condition-vs-softmax rows (final checkpoint, nats); the K-interaction rows exist only in the 250M factorial. Single-seed suites: 95% circular block-bootstrap CI over validation blocks (evaluation sampling only); suite D: mean over five seeds, paired by seed, 95% t-CI (4 df). Main effects average over the other factor; interaction $= (FWM_P - FWM_W) - (\mathrm{MM}_P - \mathrm{MM}_W)$.

Contrast (nats)	B: 124M@250M FineWebEdu [block CI]	A: 124M@2.5B FineWebEdu [block CI]	C: 1B@100M WikiText [block CI]	D: 124M@100M, 5 seeds WikiText, mean [t-CI]
MinMax–Weight K4 − softmax	+0.389 [0.385, 0.393]	+0.890 [0.881, 0.899]	+0.220 [0.214, 0.226]	+0.208 [0.191, 0.224]
MinMax–Prob K4 − softmax	+0.074 [0.072, 0.075]	+0.061 [0.060, 0.063]	+0.146 [0.142, 0.151]	+0.137 [0.129, 0.145]
FWM–Weight K4 − softmax	+0.039 [0.038, 0.040]	+0.043 [0.041, 0.045]	+0.061 [0.057, 0.065]	+0.032 [0.023, 0.040]
FWM–Prob K4 − softmax	+0.007 [0.006, 0.008]	+0.019 [0.018, 0.021]	+0.038 [0.035, 0.041]	+0.013 [0.002, 0.024]
LERP K4 − softmax	+0.052 [0.050, 0.053]	+0.005 [0.003, 0.006]	+0.096 [0.092, 0.100]	+0.098 [0.091, 0.106]
Prob − Weight, MinMax, K4	-0.316 [-0.319, -0.312]	-0.829 [-0.837, -0.820]	-0.074 [-0.078, -0.070]	-0.071 [-0.087, -0.054]
Prob − Weight, FWM, K4	-0.032 [-0.033, -0.031]	-0.023 [-0.025, -0.022]	-0.022 [-0.026, -0.019]	-0.019 [-0.028, -0.009]
FWM − MinMax, Weight-STE, K4	-0.350 [-0.354, -0.346]	-0.848 [-0.856, -0.839]	-0.159 [-0.164, -0.155]	-0.176 [-0.193, -0.158]
FWM − MinMax, Prob-STE, K4	-0.067 [-0.068, -0.065]	-0.042 [-0.044, -0.040]	-0.108 [-0.111, -0.105]	-0.124 [-0.138, -0.109]
Interaction, K4	+0.284 [0.280, 0.287]	+0.805 [0.797, 0.814]	+0.051 [0.047, 0.056]	+0.052 [0.030, 0.075]
Main effect, calibration (FWM − MinMax), K4	-0.209 [-0.211, -0.206]	-0.445 [-0.449, -0.440]	-0.134 [-0.137, -0.131]	-0.150 [-0.161, -0.138]
Main effect, surrogate (Prob − Weight), K4	-0.174 [-0.176, -0.172]	-0.426 [-0.431, -0.422]	-0.048 [-0.051, -0.045]	-0.045 [-0.052, -0.037]
FWM − MinMax, Weight-STE, K16	-0.128 [-0.129, -0.126]	-0.443 [-0.448, -0.438]	-0.056 [-0.059, -0.053]	-0.070 [-0.081, -0.058]
Main effect, calibration, K16	-0.073 [-0.074, -0.072]	—	—	-0.052 [-0.058, -0.047]
Main effect, surrogate, K16	-0.053 [-0.054, -0.052]	—	—	-0.018 [-0.025, -0.012]
Interaction, K16	+0.108 [0.107, 0.110]	—	—	+0.034 [0.017, 0.052]
$K\times$calibration (K16 − K4)	+0.135 [0.133, 0.137]	—	—	—
$K\times$surrogate (K16 − K4)	+0.121 [0.119, 0.123]	—	—	—
MinMax–Weight K4 − LERP K4	+0.337 [0.334, 0.341]	+0.886 [0.877, 0.894]	+0.124 [0.119, 0.129]	+0.109 [0.093, 0.126]
FWM–Weight K4 − LERP K4	-0.013 [-0.014, -0.012]	+0.038 [0.036, 0.040]	-0.035 [-0.039, -0.032]	-0.067 [-0.080, -0.053]
FWM–Prob K4 − LERP K4	-0.045 [-0.047, -0.043]	+0.015 [0.013, 0.016]	-0.058 [-0.061, -0.055]	-0.085 [-0.098, -0.073]

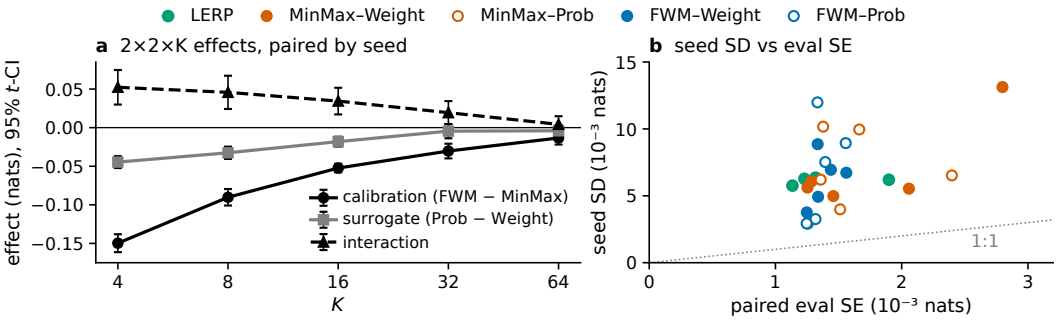


Figure S1: **124M @ 100M on five seeds, WikiText-103.** (a) Calibration, surrogate and interaction effects by K, paired by seed, 95% t intervals. (b) Seed SD of each condition's ΔNLL against the paired evaluation SE of the same contrast; every point lies above the 1:1 line.

Table S8: Five-seed $2\times2\times K$ effects in nats: mean over five paired seed-level observations (seed SD); the 95% t interval (4 df) is mean ±1.24 SD (Figure S1a), and every $K \leq 16$ interval excludes zero.

K	calibration: FWM − MinMax	surrogate: Prob − Weight	interaction
K=4	-0.1497 (0.0094)	-0.0446 (0.0062)	+0.0523 (0.0180)
K=8	-0.0902 (0.0086)	-0.0326 (0.0064)	+0.0457 (0.0173)
K=16	-0.0524 (0.0046)	-0.0182 (0.0052)	+0.0344 (0.0139)
K=32	-0.0303 (0.0077)	-0.0046 (0.0074)	+0.0194 (0.0121)
K=64	-0.0134 (0.0069)	-0.0041 (0.0058)	+0.0042 (0.0085)
pooled	-0.0672 (0.0034)	-0.0208 (0.0027)	+0.0312 (0.0047)

Table S9: Five-seed K trend: slope of ΔNLL vs softmax in nats per doubling of K (least squares against $\log_2 K$ over $K \in \{4, \ldots, 64\}$, fitted per seed); mean over seeds, 95% t-CI, seed SD.

Family	slope	95% t-CI	seed SD
LERP	-0.0222	[-0.024, -0.020]	0.0014
MinMax–Weight	-0.0472	[-0.051, -0.043]	0.0033
MinMax–Prob	-0.0301	[-0.034, -0.027]	0.0028
FWM–Weight	-0.0078	[-0.010, -0.005]	0.0020
FWM–Prob	-0.0030	[-0.007, +0.001]	0.0029

D.2 Five-seed suite

Five-seed details. Figure S1 and Tables S8–S9 give the effects and slopes; the endpoints are Table S13 in Appendix E. softmax's own endpoint ranges 4.137–4.153 across seeds. Pairwise Spearman between seeds is 0.83–0.92. The per-family slopes of ΔNLL against $\log_2 K$ are in Table S9; under FWM the gap is already small at K=4 (+0.032 Weight-STE, +0.013 Prob-STE). The cost of approximation appears between 40M and 60M tokens for the conditions that degrade; at 20M no condition is degraded (early-checkpoint contrasts pair the four seeds whose softmax run has intermediate checkpoints, the legacy seed-1337 softmax run having only its endpoint; the 100M endpoints pair all five).

D.3 Horizon and K details at 2.5B

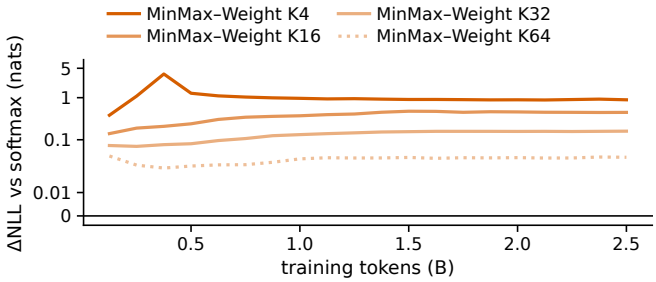


Figure S2: **MinMax–Weight K sweep along training at 2.5B**, ΔNLL vs softmax (symlog).

LERP K=1 (a single interval) is still improving at 2.5B (+0.35). FWM–Weight K=4 pays early (+0.043 at 25M, the worst K=4 condition then), because a fixed 1.5-nat grid is coarse relative to early row spans of 3–4, and then holds flat. MinMax–Weight K=4 passes through a collapse (+3.66 nats at 375M tokens, single-step gradient norms of 10^4, 98% of steps clipped over the last 1.5B tokens) and partially recovers to its +0.89 plateau (Figure S2); no Prob-STE or FWM run clips abnormally.

D.4 TAIL POLICY IN THE MINMAX → FWM CHANGE

Definition and experiment. Replacing MinMax by FWM changes the grid step near the row maximum (adaptive $(M-m)/K$ to τ/K), the dependence of the calibration and its backward on m, and the treatment of keys below the window (kept on the grid versus set to zero). The diagnostic operator *resolved* keeps FWM's step τ/K and grid origin $-\tau$ but continues the grid below $-\tau$ without bound (negative grid indices), so every key is reconstructed on the same fine grid and none is truncated; $K{=}4$ fixes the step within the window, not the number of usable levels. Two contrasts follow: MinMax→resolved, an untruncated fixed-step calibration contrast (step, its adaptivity, the m-dependence of the backward and the number of usable levels change at once), and resolved→FWM, a tail-policy contrast at a fixed grid rule. They sum to the paper's MinMax→FWM change along this path by construction, which establishes neither a unique mechanism nor a causal mediation share. Two runs (resolved × Weight-STE and × Prob-STE; Nearest, $K{=}4$, $\tau{=}6$, 124M @ 100M, WikiText-103, seed 1337, RTX 3090 / torch 2.5.1, historical STE arithmetic rather than the strict forward of Appendix C.2) were trained in code roots copied from the frozen FWM roots with a `tail_policy` field, the configuration asserted field by field against the suite-D FWM–Weight $K{=}4$ seed-1337 checkpoint; in-window unnormalized weights are bit-identical to FWM's (probabilities are not, since tail keys enter the denominator), the closed-form backward matches autograd to $4.4{\times}10^{-16}$, and evaluation follows the paper's protocol.

Result. Table S10 gives the contrasts against the suite-D seed-1337 runs. The untruncated fixed-step operator recovers most of the MinMax–FWM gap: it ends slightly below FWM under Weight-STE (-0.012 nats; path ratio 1.08, interval excluding 1) and at a similar loss under Prob-STE ($+0.001$, within the 0.01-nat single-seed margin; ratio 0.99). Zeroing the tail is therefore not required for the observed improvement in this setting. Native probability mass on tail keys is 0.008–0.010 under resolved, 0 under FWM and 0.018–0.023 under MinMax, against 0.007–0.014 softmax mass on the same keys; resolved uses 14.5 and 19.1 distinct nonzero grid levels per row against at most 5 for FWM (five nonzero levels plus the zero tail) and MinMax, so the comparison is not at equal deployment cost.

Limits. One seed (intervals cover evaluation sampling only; the calibration terms exceed the suite-D seed SD of 0.003–0.013 by an order of magnitude, the tail-policy terms do not), one K and τ, and no isolation of resolution alone or of why the fixed-step grid carries the effect; the result does not show that truncation is unnecessary elsewhere or equivalent at equal cost. A τ-floor tail variant ($z \leftarrow -\tau$) was stopped after one step, and a separate WindowFloor exploration was completed but superseded (identity and reasons in Appendix G). The two runs are supplementary.

Table S10: Tail policy in the MinMax → FWM change (hard Nearest, $K{=}4$, $\tau{=}6$, 124M @ 100M, WikiText-103, seed 1337). Paired per validation block against the suite-D seed-1337 runs; 95% circular block-bootstrap CIs (evaluation sampling only); °: $|\Delta| < 0.01$ nats, below the single-seed interpretation margin. The path ratio $(\mathrm{MinMax} - \mathrm{resolved})/(\mathrm{MinMax} - FWM)$ is descriptive and path-specific; it exceeds 1 when the tail-policy term has the opposite sign.

Surrogate	Contrast (nats)	Δ	95% CI
Weight-STE	MinMax − softmax	+0.1979	[+0.1926, +0.2031]
Weight-STE	FWM − softmax	+0.0403	[+0.0366, +0.0438]
Weight-STE	resolved − softmax	+0.0282	[+0.0246, +0.0315]
Weight-STE	MinMax − resolved (untruncated fixed-step calibration contrast)	+0.1697	[+0.1645, +0.1748]
Weight-STE	resolved − FWM (tail-policy contrast, fixed grid rule)	-0.0122	[-0.0149, -0.0096]
Weight-STE	path ratio (descriptive)	1.077	[1.061, 1.095]
Prob-STE	MinMax − softmax	+0.1399	[+0.1349, +0.1449]
Prob-STE	FWM − softmax	+0.0130	[+0.0095, +0.0159]
Prob-STE	resolved − softmax	+0.0137	[+0.0090, +0.0177]
Prob-STE	MinMax − resolved (untruncated fixed-step calibration contrast)	+0.1262	[+0.1220, +0.1309]
Prob-STE	resolved − FWM (tail-policy contrast, fixed grid rule)	+0.0007	[-0.0020, +0.0032] °
Prob-STE	path ratio (descriptive)	0.995	[0.975, 1.016]

D.5 REDUCED-PRECISION EXPONENTIAL BASELINES

Purpose and protocol. These two supplementary runs ask whether lowering the precision of the exponential, with no K-interval grid, also pretrains from scratch. They use the suite-D protocol at seed 7 (124M, WikiText-103, 100M tokens, no gradient checkpointing) on the local RTX 3090

with torch 2.5.1+cu121, the stack of the seed-7 softmax control, in a copy of the phaseB base root with the two exponential modes added, registered before training (Appendix G). Inside the attention path everything is fp32 as in every other run and the only change is how e^z is produced; there is no learned or data-dependent quantization scale beyond the row-max shift. The checkpoint configs carry the legacy field `range_grad=detach`; the reduced-precision code path reads it only for the P1 projection hook and never detaches the row maximum, which stays on the autograd path (verified in the frozen `quant_attention.py`).

Definitions. *BF16-exp*: z is cast to bfloat16, the exponential is evaluated with a bfloat16 output (input rounded before exponentiation, not only the output afterwards) and the weights are returned to fp32 for normalization; the backward is native autograd through the casts, the reduced-precision gradient of this composite map rather than a hand-written surrogate. *FP8-rounded exp (FP32 exp backward)*: $w^{\exp} = e^z$ is computed in fp32, rounded to `torch.float8_e4m3fn` and returned to fp32, $w = w^{\mathrm{fp8}} + \big(w^{\exp} - \mathrm{sg}(w^{\exp})\big)$, so the forward is the E4M3 value bit for bit (verified against an independent evaluation on CPU and GPU) while the rounding step carries an identity surrogate and the exponential keeps its fp32 derivative; the normalization Jacobian is evaluated at the rounded weights. This is a weight-level straight-through placement, not a literal "no-STE" baseline, with the fp32 exponential rather than the LERP reconstruction as surrogate. In E4M3 the exponential weights $w \in (0, 1]$ can take 56 nonzero values (49 normal values with exponents -6 to 0 and 7 subnormal multiples of 2^{-9}); round-to-nearest maps $w \le 2^{-10}$ to zero, i.e. z below about -6.93 underflows, a rounding tail rather than FWM's explicit $\tau = 6$ truncation (the exact boundary is not claimed bitwise). The counts 56 and 5 refer to reconstructed weight values, not to distinct normalized probabilities. In both modes the row maximum stays on the autograd path, so the implemented score gradient has the row-zero-sum property in exact arithmetic; the measured residual $|\sum_j \partial L/\partial s_j| / \sum_j |\partial L/\partial s_j|$ is at most 2.3×10^{-7} on random rows at three score scales on CPU and GPU (`round3/checks/zero_sum_residual_r3.json`).

Endpoints. Table S11 gives the 100M endpoints against the same-seed softmax, all evaluated on one RTX 3090 (243 blocks, batch 1, TF32 off), with 95% circular moving-block bootstrap intervals (block 16, 4000 resamples) over validation blocks; each row is one run. The new runs end -0.0034 [-0.0058, -0.0009] (BF16-exp) and $+0.0028$ [$+0.0002$, $+0.0055$] (FP8-rounded exp) nats from softmax: small differences that the evaluation intervals detect, both below 0.01 nats in magnitude. The training stack differs across rows and same-GPU evaluation does not remove that, so the historical and strict FWM–Prob rows are background (historical FWM–Prob $K{=}4$ five-seed mean $+0.0132$, SD 0.0089, a seed-level spread, not an evaluation CI); the same-seed LERP and MinMax rows of the table are NLL gaps at the same scale, not all of them instabilities.

Table S11: Reduced-precision exponential baselines and same-seed references at 100M tokens (124M, WikiText-103, seed 7): ΔNLL against the seed-7 softmax run, paired per validation block, all evaluated on one RTX 3090 (243 blocks); 95% circular block-bootstrap CI over validation blocks (evaluation sampling only). Training GPU and software stack per row from the run manifests and logs; "author-confirmed" rows are cloud-pod runs whose logs did not record the GPU model. Historical FWM–Prob $K{=}4$ five-seed mean $+0.0132$ (SD 0.0089) is seed-level background. Data: `round3/reports/r3_lowprec_contrasts.csv`.

Operator (seed 7, 124M @ 100M)	Training GPU; software stack	ΔNLL	95% eval. CI
BF16-exp (new)	local RTX 3090; torch 2.5.1+cu121	-0.0034	[-0.0058, -0.0009]
FP8-rounded exp, FP32 exp backward (new)	local RTX 3090; torch 2.5.1+cu121	+0.0028	[+0.0002, +0.0055]
LERP $K{=}32$	local RTX 3090; torch 2.5.1+cu121	+0.0047	[+0.0024, +0.0069]
LERP $K{=}4$	local RTX 3090; torch 2.5.1+cu121	+0.0942	[+0.0913, +0.0969]
MinMax–Prob $K{=}4$ (historical)	RTX 5090 pod (author-confirmed); torch not recorded	+0.1452	[+0.1412, +0.1493]
MinMax–Weight $K{=}4$ (historical)	local RTX 3090; torch 2.5.1+cu121	+0.2245	[+0.2190, +0.2302]
FWM–Prob $K{=}4$ (historical)	RTX 5090 pod (author-confirmed); torch not recorded	+0.0083	[+0.0057, +0.0107]
FWM–Prob $K{=}4$ (strict forward, App. C.2)	RTX 5090 (host7); torch 2.13.0+cu132	-0.0036	[-0.0070, -0.0003]

Trajectories and stability. Figure S3 shows the differences from the same-seed softmax run at the five evaluated checkpoints (checkpoint values, not a continuous bound): both new runs stay within ±0.011 nats of softmax throughout, while LERP $K{=}4$ opens its gap between 40M and 60M. The training log (one point per 5M tokens) shows no nonfinite values or excess loss increases (largest train-loss increase between consecutive logs 0.035 for BF16-exp, 0.034 for FP8-rounded exp and 0.034 for softmax), which does not exclude excursions between logs. Final pre-clip gradient norms

are 0.497 / 0.482 / 0.498 and the 100M score geometry (layer mean over 8 validation blocks) is close to softmax: row span 25.9 / 24.9 / 26.2, softmax entropy 3.75 / 3.75 / 3.74, query and key norms within 3% (BF16-exp / FP8-rounded exp / softmax); LERP $K{=}4$ has span 13.2 at the same point.

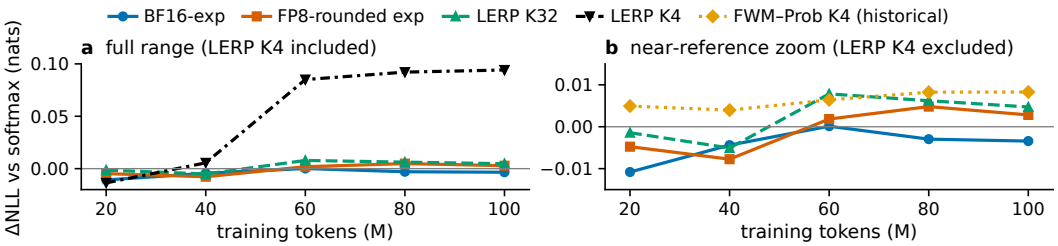


Figure S3: **Reduced-precision exponential baselines** (suite-D protocol, 124M, WikiText-103, seed 7): validation NLL minus the same-seed softmax run at the five evaluated checkpoints, all evaluated on one RTX 3090. (a) Full range including LERP $K{=}4$; (b) near-reference trajectories, LERP $K{=}4$ excluded, with the historical FWM–Prob $K{=}4$ run (different training stack). One run per curve; endpoint evaluation intervals in Table S11.

Native FP8 backward diagnostic. With PyTorch's native backward for the cast, the gradient itself is cast to E4M3. On the seed-7 initialization and the first training update (128 micro-batches, training precision, unscaled), this flushed the Q/K projection parameter gradients to exactly zero (L2 norm 0, all gradients finite); with the fp32 exponential backward used for training they are ordinary (0.0485; `round3/checks/diag_native_fp8.json`). This is a diagnostic of that unscaled implementation at initialization; it does not show that FP8 training is impossible or that fp32 exponential gradients are necessary in general (gradient scaling and other backward designs were not tested).

Reading and limits. Like FWM and unlike MinMax, both baselines use a fixed quantization/reconstruction rule in row-max-shifted coordinates, independent of the learned row span, and their rounding does not coarsen as the span grows (floating-point formats are not equally spaced grids; BF16-exp additionally rounds the input z), although the realized mismatch still depends on the score distribution. This is consistent with the range–resolution reading of §5.4 but not a controlled test of it: format, grid density, tail treatment and backward rule all differ from the K-interval operators. The controls show that the tested reductions in exponential precision remain close to the fp32-softmax reference at this horizon; they establish neither equivalence (no equivalence test was run; the downstream ± 0.01-nat margin is not applied here), nor a cross-seed ranking of BF16-exp, FP8-rounded exp, FWM–Prob $K{=}4$ and softmax against training randomness (softmax's own five-seed endpoint SD is 0.0062), nor a cost-matched comparison with five-level reconstruction (representable values are not a proxy for compute, storage width or speed); one seed, 100M tokens, no longer horizon or coarser format (FP4, 4-bit lookup). The calibration and surrogate interventions elsewhere provide the controlled evidence for the large failures within the studied operator family.

D.6 Score geometry along training at 2.5B

Additional geometry observations. At 2.5B (Figure S4) MinMax–Weight $K{=}4$ peaks at span 287 and $K{=}16$ reaches 191; the largest logit dispersion reported for Gemma 7B is about 33 (Veličković et al., 2024), so softmax's 25–32 is in that range while MinMax–Weight's and the detached runs' spans are not. The softmax mass in MinMax–Weight $K{=}4$'s top rounding bucket reaches 99.5%: the operator maps almost all mass to one grid value, which is not the model concentrating probability on one key. FWM is structurally the calibration-side counterpart of the explicit clip that binarized networks needed (Hubara et al., 2016), an analogy not promoted to a shared mechanism. Within-run Spearman between span and the NLL gap over the 10 checkpoints of the 250M MinMax–Weight run is about 0.9. Each link of the range–resolution feedback hypothesis has a literature counterpart (Veličković et al., 2024; Hoffmann et al., 2024); the tail-policy comparison of Appendix D.4 separates one part of the compound calibration change, and freezing h mid-training, a surrogate swap from a saved checkpoint, QK-normalization (Dehghani et al., 2023) or a learning-rate sweep (Wortsman et al., 2024) would separate the rest; none was run. Geometric diagnostics moved before the loss

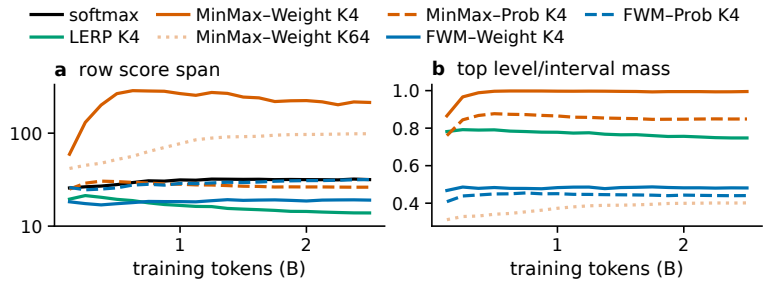


Figure S4: **Layer-mean score geometry along training at 2.5B**, $K{=}4$ conditions, softmax and MinMax–Weight $K{=}64$ (dotted), on each model's own scores over 8 validation blocks. (a) Row span; (b) softmax mass on the keys in the operator's top set, which is the top rounding bucket for Nearest and the top interval for LERP (these sets differ, so the panel compares each operator with its own top set). The native interval width h and the per-row TV of the same six approximate conditions and checkpoints are Figure 5a,b.

in the detach failure but after it in the formal MinMax–Weight runs (Figure S5), so they are not a universal early warning; across the five-seed conditions the endpoint gap tracks h and span only as a cross-condition association (Figure S6). All associations in this appendix are descriptive.

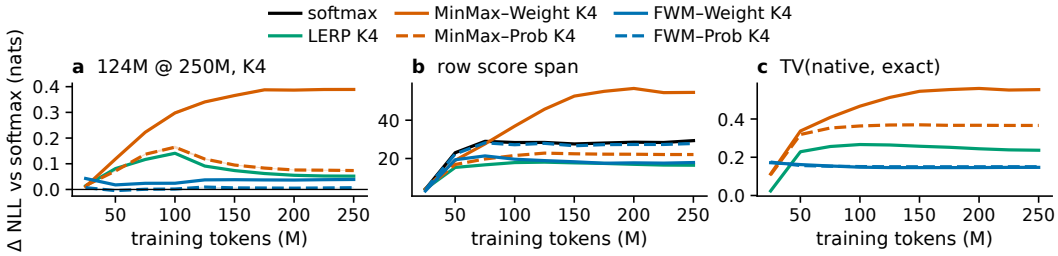


Figure S5: **124M @ 250M, $K{=}4$ conditions.** (a) ΔNLL vs softmax with 95% CI at 25M resolution. (b) Row span and (c) TV along training. At 50M the MinMax–Weight/MinMax–Prob NLL gap (0.047) is open while their geometry is nearly identical.

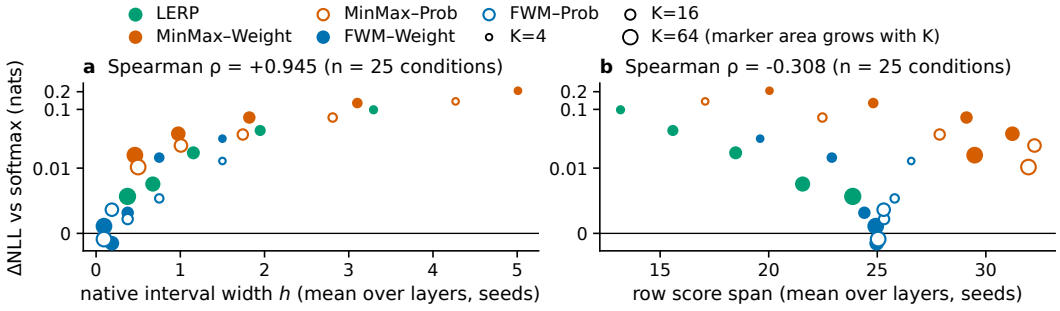


Figure S6: **Five-seed condition means: geometry vs endpoint gap.** (a) Native interval width h and (b) row score span, each averaged over layers and seeds at the 100M endpoint, against the condition's mean ΔNLL vs softmax (symlog axis). Spearman correlations are across the 25 approximate conditions of a designed grid: cross-condition associations, not causal estimates.

E Full endpoint results and downstream evaluation

E.1 Downstream evaluation of the 2.5B suite

Pre-declared condition groups. Before any downstream run the fifteen 2.5B endpoints were assigned to three groups by their 2.5B ΔNLL vs softmax, with the declared expectation in quotes:

Table S12: All single-seed endpoints (final checkpoint, full validation, 95% circular block-bootstrap CI over validation blocks). NLL is the mean over 1024-token blocks of the per-token NLL under GPT-2 BPE (nats/token); the corresponding perplexity $\exp(\text{mean NLL})$ is *token-level* and not comparable to word-level WikiText-103 perplexities in the literature.

Condition	NLL	Δ vs softmax	95% block CI
124M @ 250M, FineWebEdu-3B, 976 blocks			
softmax	3.8883	—	—
LERP K4	3.9401	+0.0518	[+0.050, +0.053]
MinMax–Weight K4	4.2774	+0.3892	[+0.385, +0.393]
MinMax–Weight K16	4.0127	+0.1244	[+0.123, +0.126]
MinMax–Prob K4	3.9618	+0.0735	[+0.072, +0.075]
MinMax–Prob K16	3.9054	+0.0171	[+0.016, +0.018]
FWM–Weight K4	3.9271	+0.0388	[+0.038, +0.040]
FWM–Weight K16	3.8851	-0.0032	[-0.004, -0.002]
FWM–Prob K4	3.8951	+0.0068	[+0.006, +0.008]
FWM–Prob K16	3.8862	-0.0020	[-0.003, -0.001]
1B @ 100M, WikiText-103, 243 blocks			
softmax	4.0904	—	—
LERP K4	4.1867	+0.0963	[+0.092, +0.100]
LERP K16	4.1322	+0.0417	[+0.039, +0.044]
LERP K32	4.1101	+0.0197	[+0.017, +0.022]
MinMax–Weight K4	4.3107	+0.2202	[+0.214, +0.226]
MinMax–Weight K16	4.1653	+0.0748	[+0.072, +0.077]
MinMax–Weight K32	4.1357	+0.0453	[+0.043, +0.048]
MinMax–Prob K4	4.2368	+0.1464	[+0.142, +0.151]
FWM–Weight K4	4.1512	+0.0608	[+0.057, +0.065]
FWM–Weight K16	4.1090	+0.0186	[+0.016, +0.021]
FWM–Weight K32	4.1067	+0.0163	[+0.014, +0.019]
FWM–Prob K4	4.1288	+0.0383	[+0.035, +0.041]

Condition	NLL	Δ vs softmax	95% block CI
124M @ 2.5B, FineWebEdu-3B, 976 blocks			
softmax	3.1928	—	—
LERP K1	3.5386	+0.3458	[+0.340, +0.351]
LERP K2	3.2025	+0.0097	[+0.008, +0.011]
LERP K4	3.1975	+0.0047	[+0.003, +0.006]
LERP K16	3.1931	+0.0003	[-0.001, +0.002]
LERP K32	3.1947	+0.0019	[+0.000, +0.003]
MinMax–Weight K4	4.0831	+0.8903	[+0.881, +0.899]
MinMax–Weight K16	3.6397	+0.4469	[+0.442, +0.452]
MinMax–Weight K32	3.3527	+0.1600	[+0.158, +0.162]
MinMax–Weight K64	3.2311	+0.0383	[+0.037, +0.040]
MinMax–Prob K4	3.2542	+0.0614	[+0.060, +0.063]
FWM–Weight K2	3.3226	+0.1298	[+0.128, +0.132]
FWM–Weight K4	3.2355	+0.0427	[+0.041, +0.045]
FWM–Weight K16	3.1967	+0.0039	[+0.002, +0.005]
FWM–Prob K4	3.2121	+0.0193	[+0.018, +0.021]

Table S13: Five-seed endpoints (124M @ 100M, WikiText-103; mean NLL in nats/token, GPT-2 BPE; the token-level perplexity $\exp(\text{mean NLL})$ is not comparable to word-level perplexities). Δ is paired by seed; the t interval has 4 df; high-K FWM cells are within seed noise of softmax and are not strictly ordered.

Condition	mean NLL	seed SD	Δ vs softmax	95% t-CI	seed SD of Δ	paired eval SE
softmax	4.1457	0.0062	—	—	—	—
LERP K4	4.2441	0.0080	+0.0985	[+0.091, +0.106]	0.0062	0.0019
LERP K8	4.1894	0.0098	+0.0437	[+0.036, +0.052]	0.0064	0.0013
LERP K16	4.1639	0.0074	+0.0182	[+0.015, +0.022]	0.0029	0.0013
LERP K32	4.1532	0.0089	+0.0076	[-0.000, +0.015]	0.0063	0.0012
LERP K64	4.1513	0.0098	+0.0057	[-0.001, +0.013]	0.0058	0.0011
MinMax–Weight K4	4.3532	0.0097	+0.2076	[+0.191, +0.224]	0.0131	0.0028
MinMax–Weight K8	4.2738	0.0092	+0.1282	[+0.121, +0.135]	0.0055	0.0021
MinMax–Weight K16	4.2184	0.0104	+0.0727	[+0.067, +0.079]	0.0050	0.0015
MinMax–Weight K32	4.1842	0.0100	+0.0385	[+0.032, +0.045]	0.0056	0.0013
MinMax–Weight K64	4.1623	0.0066	+0.0166	[+0.009, +0.024]	0.0061	0.0013
MinMax–Prob K4	4.2825	0.0080	+0.1368	[+0.129, +0.145]	0.0065	0.0024
MinMax–Prob K8	4.2184	0.0120	+0.0727	[+0.060, +0.085]	0.0100	0.0017
MinMax–Prob K16	4.1830	0.0064	+0.0374	[+0.030, +0.045]	0.0062	0.0014
MinMax–Prob K32	4.1699	0.0054	+0.0243	[+0.019, +0.029]	0.0040	0.0015
MinMax–Prob K64	4.1561	0.0125	+0.0104	[-0.002, +0.023]	0.0102	0.0014
FWM–Weight K4	4.1774	0.0106	+0.0317	[+0.023, +0.040]	0.0067	0.0016
FWM–Weight K8	4.1608	0.0142	+0.0151	[+0.004, +0.026]	0.0089	0.0013
FWM–Weight K16	4.1488	0.0089	+0.0032	[-0.005, +0.012]	0.0069	0.0014
FWM–Weight K32	4.1442	0.0090	-0.0015	[-0.008, +0.005]	0.0049	0.0013
FWM–Weight K64	4.1468	0.0078	+0.0011	[-0.004, +0.006]	0.0038	0.0012
FWM–Prob K4	4.1589	0.0133	+0.0132	[+0.002, +0.024]	0.0089	0.0016
FWM–Prob K8	4.1510	0.0052	+0.0054	[+0.002, +0.009]	0.0029	0.0012
FWM–Prob K16	4.1478	0.0052	+0.0022	[-0.002, +0.006]	0.0033	0.0013
FWM–Prob K32	4.1493	0.0146	+0.0036	[-0.011, +0.019]	0.0120	0.0013
FWM–Prob K64	4.1448	0.0122	-0.0009	[-0.010, +0.008]	0.0075	0.0014

large-gap, ΔNLL 0.13–0.89, MinMax–Weight K4 (+0.890), MinMax–Weight K16 (+0.447), LERP K1 (+0.346), MinMax–Weight K32 (+0.160), FWM–Weight K2 (+0.130), "should be clearly detectable"; *intermediate-gap*, 0.02–0.06, MinMax–Prob K4 (+0.061), FWM–Weight K4 (+0.043), MinMax–Weight K64 (+0.038), FWM–Prob K4 (+0.019), "possibly detectable, task dependent"; *near-baseline*, < 0.01, LERP K2 (+0.010), LERP K4 (+0.005), FWM–Weight K16 (+0.004), LERP K32 (+0.002), LERP K16 (+0.0003), "expected not detectable". The lists are the pre-declaration (recorded as pathological / intermediate / good, renamed without changing membership); the ranges are descriptive and three members sit at their rounded edges. They fix the denominators 35, 28 and 35 of §5.5.

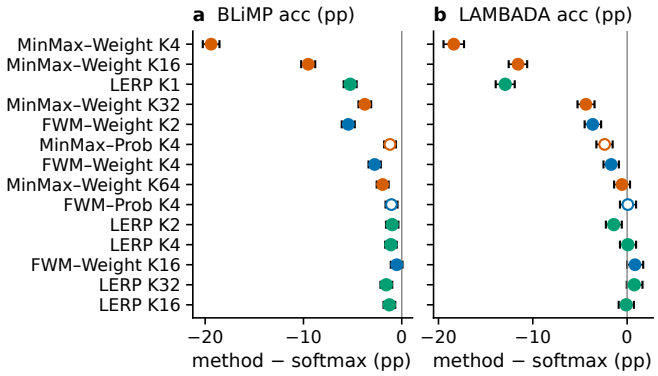


Figure S7: **Downstream accuracy of the fourteen approximate 2.5B endpoints**, method minus softmax in percentage points with 95% paired bootstrap CIs: (a) BLiMP macro accuracy, (b) LAMBADA accuracy. Rows are ordered by 2.5B ΔNLL; the ΔNLL values on the training domain and the three held-out corpora are in Tables S12 and S15.

Held-out corpora (layer 1). On WikiText-103, PTB and C4 the Spearman correlation with the in-domain ordering is 0.996, 0.952 and 0.996; MinMax–Weight $K{=}4$ goes from $+0.89$ in domain to $+1.22$ on WikiText-103 and $+1.12$ on PTB, unchanged on C4. Table S14 gives the paired 95% intervals for the five near-baseline conditions on the three corpora (counts under P2 below): the differences are below 0.06 nats and of inconsistent sign (the four LERP conditions *below* softmax on PTB, FWM–Weight $K{=}16$ above it on PTB and C4), small but detectable held-out differences in both directions, neither "no detectable difference" nor equivalence. Table S15 separates the native-forward and softmax-forward evaluations of the same weights.

Table S14: Layer 1, near-baseline group: paired ΔNLL vs softmax on the held-out corpora (native forward) with 95% CIs (circular moving-block bootstrap, block 16, for WikiText-103 and PTB; document-cluster bootstrap for C4; 4000 resamples), BH q within each corpus's 14-comparison family, and the two independent labels of Appendix C.

Condition	Corpus	ΔNLL	95% CI	BH q	CI excl. 0	CI inside ±0.01
LERP K2	WikiText-103 test	+0.0052	[-0.0005, +0.0106]	0.087	no	no
LERP K2	PTB test	-0.0162	[-0.0235, -0.0090]	<0.001	yes	no
LERP K2	C4 validation	+0.0131	[+0.0096, +0.0165]	<0.001	yes	no
LERP K4	WikiText-103 test	+0.0023	[-0.0020, +0.0068]	0.318	no	yes
LERP K4	PTB test	-0.0177	[-0.0238, -0.0123]	<0.001	yes	no
LERP K4	C4 validation	+0.0045	[+0.0012, +0.0078]	0.007	yes	yes
LERP K16	WikiText-103 test	-0.0084	[-0.0130, -0.0036]	<0.001	yes	no
LERP K16	PTB test	-0.0578	[-0.0639, -0.0519]	<0.001	yes	no
LERP K16	C4 validation	-0.0007	[-0.0036, +0.0022]	0.656	no	yes
LERP K32	WikiText-103 test	-0.0059	[-0.0099, -0.0021]	0.004	yes	yes
LERP K32	PTB test	-0.0393	[-0.0460, -0.0330]	<0.001	yes	no
LERP K32	C4 validation	+0.0011	[-0.0015, +0.0038]	0.444	no	yes
FWM–Weight K16	WikiText-103 test	-0.0021	[-0.0070, +0.0026]	0.391	no	yes
FWM–Weight K16	PTB test	+0.0227	[+0.0167, +0.0286]	<0.001	yes	no
FWM–Weight K16	C4 validation	+0.0050	[+0.0018, +0.0082]	0.002	yes	yes

Benchmarks (layer 3). The seven stage-C configurations are six benchmarks, SciQ scored with and without its support passage (Table S16, symbols as in Figure 6; BLiMP and LAMBADA with CIs in Figure S7). BLiMP is the macro average over 67 paradigms of 200 minimal pairs each, resampled by item (per-item rows total 1.4M across conditions, a bookkeeping count). The intermediate-gap group separates from softmax only on BLiMP and LAMBADA (6 of 28 comparisons); FWM–Weight $K{=}16$ does not separate on BLiMP (-0.5 pp, $q = 0.10$); the mean near-baseline difference ($+0.3$ pp) is descriptive only. No condition loses more of the SciQ support-passage benefit than softmax after correction.

Induction and retrieval probes (layer 2, probes A and B). Table S17 reports every evaluated condition on probe A (400 families $\times$ 5 distances $=$ 2,000 induction samples, each with a matched control) and probe B (2,400 samples), native forward, against the softmax reference: per-token target NLL, the induction gain (control minus induction target NLL, probe A) and top-1 accuracy. The NLL columns carry paired bootstrap CIs. The five distances of a probe-A family share one base sequence

Table S15: Layer 1: ΔNLL vs softmax on three held-out corpora with the model's native forward, with softmax substituted at evaluation on the same weights, and their difference (softmax-fwd − native). The softmax reference in each column is the softmax model evaluated in the same mode. In 37 of the 42 approximate-condition cells the substituted gap exceeds the native one.

	WikiText-103 test			PTB test			C4 validation		
Condition	native	softmax-fwd	diff	native	softmax-fwd	diff	native	softmax-fwd	diff
MinMax–Weight K4	+1.215	+3.549	+2.334	+1.122	+3.309	+2.187	+0.865	+2.942	+2.077
MinMax–Weight K16	+0.588	+1.232	+0.644	+0.526	+0.905	+0.380	+0.435	+0.971	+0.536
LERP K1	+0.652	+4.947	+4.295	+0.702	+4.735	+4.034	+0.390	+4.637	+4.247
MinMax–Weight K32	+0.209	+0.242	+0.033	+0.176	+0.202	+0.026	+0.156	+0.210	+0.054
FWM–Weight K2	+0.180	+0.288	+0.108	+0.135	+0.236	+0.101	+0.128	+0.245	+0.117
MinMax–Prob K4	+0.097	+0.208	+0.111	+0.091	+0.173	+0.082	+0.067	+0.143	+0.077
FWM–Weight K4	+0.066	+0.084	+0.019	+0.049	+0.057	+0.008	+0.044	+0.046	+0.002
MinMax–Weight K64	+0.045	+0.057	+0.013	+0.068	+0.063	-0.005	+0.038	+0.048	+0.010
FWM–Prob K4	+0.027	+0.025	-0.001	-0.020	-0.026	-0.006	+0.021	+0.015	-0.007
LERP K2	+0.005	+1.814	+1.809	-0.016	+1.670	+1.686	+0.013	+1.685	+1.672
LERP K4	+0.002	+0.174	+0.172	-0.018	+0.169	+0.187	+0.004	+0.168	+0.163
FWM–Weight K16	-0.002	+0.002	+0.004	+0.023	+0.030	+0.007	+0.005	+0.004	-0.001
LERP K32	-0.006	-0.005	+0.001	-0.039	-0.036	+0.003	+0.001	+0.003	+0.002
LERP K16	-0.008	-0.002	+0.007	-0.058	-0.052	+0.006	-0.001	+0.006	+0.007

Table S16: Layer 3, all 14 conditions: accuracy minus softmax in percentage points, paired per item and computed from the per-item differences (not from rounded absolutes); the header row gives softmax's absolute accuracy. BLiMP is the macro-average over 67 paradigms (200 pairs each); HellaSwag/PIQA/ARC-Easy use `acc_norm`; LAMBADA and SciQ use `acc`; SciQ-ns is without the support passage. Rows are ordered by 2.5B ΔNLL with rules between the large-gap, intermediate-gap and near-baseline groups. *: BH $q < 0.05$ within the configuration×metric family of 14 comparisons (the symbol of Figure 6); $^\circ$: 95% paired CI excludes zero but $q \geq 0.05$. Per-cell CIs: `stageC/tables/paired_comparisons.csv`; absolute accuracies per condition: file `tab_downstream_stageC_absolute.tex` in `compact/migrated/`.

Condition	BLiMP	HellaSwag	PIQA	ARC-E	LAMBADA	SciQ	SciQ-ns
softmax (absolute, %)	79.5	30.0	59.7	45.2	21.6	73.1	51.2
MinMax–Weight K4	-19.4*	-3.9*	-4.1*	-9.6*	-18.4*	-19.3*	-17.6*
MinMax–Weight K16	-9.5*	-2.7*	-2.2°	-5.7*	-11.6*	-9.3*	-8.7*
LERP K1	-5.2*	-1.9*	-0.2	-3.9*	-12.9*	-8.5*	-5.2*
MinMax–Weight K32	-3.8*	-1.6*	-0.4	-3.5*	-4.4*	-2.2	-1.4
FWM–Weight K2	-5.4*	-1.1*	-1.0	-2.6*	-3.6*	-1.2	-1.4
MinMax–Prob K4	-1.2*	-0.6°	+1.3	-1.2	-2.4*	-0.5	+0.9
FWM–Weight K4	-2.8*	-0.5	+1.0	-0.2	-1.7*	+1.6	+0.9
MinMax–Weight K64	-1.9*	-0.3	+0.4	-0.7	-0.5	+1.1	+0.6
FWM–Prob K4	-1.1*	-0.4	+0.8	-0.8	+0.1	-0.7	+2.3
LERP K2	-1.0*	-0.0	+0.2	-2.4*	-1.4*	+2.0	+1.5
LERP K4	-1.1*	+0.4	-0.2	-0.6	+0.1	+1.3	+1.5
FWM–Weight K16	-0.5	-0.1	+1.7°	-0.3	+0.8	+3.9*	+3.3*
LERP K32	-1.6*	+0.2	+0.5	-0.6	+0.8	+1.0	+0.4
LERP K16	-1.3*	+0.2	+0.8	+0.3	-0.1	+0.4	+0.5

and one (A, B) pair, so the probe-A intervals resample the 400 families, keeping every distance, the induction/control pairing and the method pairing inside a family; probe-B samples are independent and are resampled directly. (A sample-level resample of the 2,000 items, used earlier, gave narrower intervals, largest half-width 0.12 against 0.21 nats; two induction-gain cells, LERP $K{=}2$ and MinMax–Weight $K{=}16$, now include zero; no target-NLL cell changes; top-1 columns are point estimates.) Per-configuration results are in `downstream_v1/results/layer2/probe_scores.csv`. The reference model itself barely solves either probe (top-1 0.05% on A, 0.7% on B; its induction gain falls from 1.22 nats at distance 64 to 0.21 at 896, 17%), so differences between conditions are measured at a baseline with little of the ability the probes target: P4 (degradation growing with distance) could not be tested, and the large, detectable target-NLL differences of both signs on A and B are reported as probe outcomes rather than as evidence; the largest, MinMax–Prob $K{=}4$'s -1.28-nat target NLL and $+1.47$-nat induction gain on probe A (top-1 from 0.05% to 4%), is not interpreted as long-range copying. This is a limit of the reference model at 124M, not a null result and not equivalence between conditions.

Copying probe (layer 2, probe C). Table S18 gives the per-token target-NLL difference from softmax by copied length with softmax's absolute target NLL. The slope against $\log_2 L$ is steeper

Table S17: Probes A and B, all 14 conditions plus the softmax reference (124M @ 2.5B endpoints, native forward). Softmax row: absolute values; other rows: difference from softmax. The target-NLL and induction-gain columns carry 95% paired-bootstrap CIs (4000 resamples; probe A resampled by family over the 400 families, probe B by sample); the top-1 columns are point estimates only, in accuracy units (fraction). Induction gain: target NLL without minus with the earlier occurrence (probe A). Data: `downstream_v1/results/layer2/probe_scores.csv` (config `all`) and `probe_scores_A_all_familyboot_20260925.csv`.

	Probe A: induction (n=2,000 induction samples)			Probe B: retrieval (n=2,400)	
Condition	target NLL	induction gain	top-1	target NLL	top-1
softmax (absolute)	11.160	0.865	0.0005	10.255	0.0071
MinMax–Weight K4	+0.583 [+0.377, +0.775]	-0.515 [-0.589, -0.440]	-0.0005	+1.708 [+1.622, +1.799]	-0.0071
MinMax–Weight K16	+0.321 [+0.134, +0.496]	-0.087 [-0.174, +0.003]	+0.0060	+1.298 [+1.217, +1.383]	-0.0067
LERP K1	+1.212 [+1.019, +1.399]	-0.753 [-0.825, -0.680]	-0.0005	+1.508 [+1.432, +1.587]	-0.0067
MinMax–Weight K32	+0.214 [+0.047, +0.378]	-0.170 [-0.237, -0.103]	+0.0005	+1.716 [+1.638, +1.798]	-0.0058
FWM–Weight K2	-0.059 [-0.245, +0.119]	+0.168 [+0.084, +0.259]	+0.0045	+0.811 [+0.738, +0.883]	-0.0050
MinMax–Prob K4	-1.281 [-1.496, -1.067]	+1.469 [+1.300, +1.638]	+0.0400	+0.658 [+0.579, +0.736]	-0.0004
FWM–Weight K4	-0.070 [-0.232, +0.098]	+0.116 [+0.040, +0.197]	+0.0020	+0.816 [+0.743, +0.892]	-0.0058
MinMax–Weight K64	+1.204 [+1.007, +1.394]	-0.601 [-0.675, -0.526]	-0.0005	+0.628 [+0.554, +0.697]	-0.0021
FWM–Prob K4	-0.559 [-0.725, -0.395]	+0.723 [+0.627, +0.822]	+0.0100	+0.772 [+0.706, +0.840]	-0.0054
LERP K2	+0.234 [+0.052, +0.405]	-0.068 [-0.141, +0.008]	+0.0005	+0.541 [+0.464, +0.621]	-0.0050
LERP K4	-0.219 [-0.390, -0.045]	+0.093 [+0.008, +0.184]	+0.0010	+0.771 [+0.696, +0.848]	-0.0050
FWM–Weight K16	+0.737 [+0.579, +0.897]	-0.351 [-0.406, -0.292]	+0.0000	-0.039 [-0.114, +0.035]	+0.0046
LERP K32	-0.170 [-0.330, -0.019]	+0.278 [+0.198, +0.360]	+0.0010	+0.725 [+0.655, +0.794]	-0.0037
LERP K16	-0.310 [-0.485, -0.132]	+0.471 [+0.365, +0.581]	+0.0025	+0.473 [+0.404, +0.539]	-0.0029

under Prob-STE than Weight-STE at matched calibration and K=4 (MinMax 0.94 vs 0.11, FWM 0.88 vs 0.23); under MinMax Prob-STE is lower at every length, under FWM lower at L=32, 64 and higher at L=128. The softmax-forward substitution was run on the three probes for six conditions, not for the benchmark battery.

Table S18: Layer 2, probe C (verbatim sequence copying): softmax's absolute per-token target NLL, then each condition's per-token target-NLL difference from softmax by copied length L and the slope of that difference against $\log_2 L$ [95% CI].

Condition	Δ at L=32	L=64	L=128	slope vs $\log_2 L$ [95% CI]
softmax, absolute target NLL	1.723	1.576	1.776	—
MinMax–Weight K4	+7.13	+7.06	+7.34	+0.106 [+0.064, +0.146]
MinMax–Weight K16	+1.91	+1.48	+1.24	-0.337 [-0.375, -0.300]
LERP K1	+7.96	+9.24	+9.89	+0.963 [+0.929, +0.997]
MinMax–Weight K32	+0.50	+0.40	+0.26	-0.119 [-0.145, -0.095]
FWM–Weight K2	+3.21	+3.91	+4.67	+0.727 [+0.684, +0.770]
MinMax–Prob K4	+0.82	+1.41	+2.69	+0.939 [+0.897, +0.980]
FWM–Weight K4	+0.54	+0.70	+1.01	+0.234 [+0.200, +0.268]
MinMax–Weight K64	+0.24	+0.42	+0.36	+0.057 [+0.028, +0.085]
FWM–Prob K4	+0.18	+0.48	+1.94	+0.877 [+0.847, +0.907]
LERP K2	+0.37	+0.30	+0.70	+0.163 [+0.134, +0.190]
LERP K4	+0.07	+0.05	+0.47	+0.200 [+0.174, +0.226]
FWM–Weight K16	-0.00	-0.05	-0.07	-0.033 [-0.056, -0.011]
LERP K32	-0.08	+0.05	+0.27	+0.178 [+0.154, +0.201]
LERP K16	-0.15	-0.01	+0.22	+0.185 [+0.161, +0.208]

Pre-declared predictions, outcome. P1 (large-gap group detectable on layers 2 and 3): supported. P2 (near-baseline group not detectable): not supported on held-out LM NLL, where 10 of the 15 paired intervals exclude zero (Table S14) although 7 lie inside the ± 0.01-nat margin, nor uniformly on the benchmarks, where 8 of the group's 35 primary-accuracy comparisons reach $q < 0.05$ (six negative, on BLiMP, ARC-Easy and LAMBADA; two positive, on the two SciQ configurations; §5.5). P3 (probe degradation exceeds LM degradation): supported on probe C only (13 of 14 conditions). P4 (degradation grows with retrieval distance): not testable, because the reference model's own induction gain decays to 17% by distance 896. P5 (softmax-forward reversal larger on probes than on LM): holds in 5 of the 18 cells of the substitution on probes A/B/C for six conditions, so it fails as measured, and it is not an effect-size ratio, because LM and probe metrics average over different token distributions, positions, tasks and sample sizes. A stage-B assumption that downstream pairing would shrink variance like corpus NLL ($s \approx 0.20$) was contradicted by the realized shrinkage (0.27–1.32, median 0.64), which is why every interval is built from realized per-item differences.

F THE DETACHED-GRADIENT RECORD AND ITS REPLICATION

The legacy `quant_attention.py` computed `row_max = att.masked_fill(~valid, -inf).amax(-1, keepdim=True).detach()` and the analogous `row_min`; the later implementation adds `-range_grad {detach,full}` on a shared forward code path, so forward values are the same. The historical detached runs are 100M-token runs on seed 1337 (the legacy softmax run, counted in suite D; Nearest $K{=}2/4/8/16$, LERP $K{=}4/32$) and six 5M-token confirmatory Nearest runs ($K{=}2/4/8$ on seeds 1338 and 1339) with 5M softmax references on seeds 1337, 1338 and 1339, all WikiText-103, 124M, with the optimizer settings of the later 100M runs (the seed-1337 5M Nearest runs belong to a pre-phaseB pilot and are not part of this record). They are excluded from every formal table and kept for provenance and for the P1 result. Table S19 gives the standardized endpoints, including the full-gradient endpoints against softmax, and Figure S8 the MinMax–Weight trajectories.

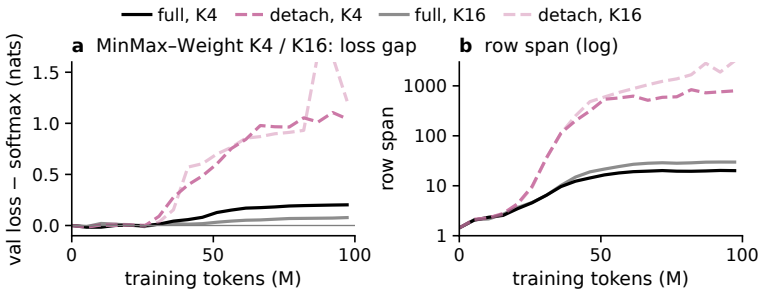


Figure S8: **Detached vs full calibration gradient, Nearest, seed 1337 (124M @ 100M, WikiText-103).** MinMax–Weight $K{=}4$ (solid colour) and $K{=}16$ (faded), detached and full backward: (a) logged validation loss minus the seed-1337 softmax run and (b) logged layer-mean row span along training. Endpoints with CIs: Table S19.

Table S19: Detached vs full calibration gradient at 100M tokens, paired on 243 WikiText-103 validation blocks (95% block-bootstrap CIs). Softmax (seed 1337) NLL 4.1499. The full-gradient runs are the seed-1337 members of suite D. No full-gradient $K{=}2$ run exists.

Condition (seed 1337, 100M)	historical backward	historical − softmax	full-gradient − softmax	historical − full
MinMax–Weight K2	detach	+2.467 [+2.410, +2.525]	—	—
MinMax–Weight K4	detach	+1.023 [+1.000, +1.045]	+0.1979 [+0.193, +0.203]	+0.825 [+0.806, +0.845]
MinMax–Weight K8	detach	+3.200 [+3.141, +3.258]	+0.1331 [+0.130, +0.137]	+3.066 [+3.009, +3.125]
MinMax–Weight K16	detach	+1.157 [+1.131, +1.185]	+0.0778 [+0.076, +0.080]	+1.079 [+1.053, +1.107]
LERP K4	detach	+0.748 [+0.733, +0.764]	+0.0944 [+0.091, +0.098]	+0.654 [+0.639, +0.668]
LERP K32	detach	+0.736 [+0.719, +0.752]	-0.0008 [-0.004, +0.002]	+0.737 [+0.721, +0.752]
LERP K32 (P1)	detach + zero-sum proj.	+0.996 [+0.977, +1.013]	-0.0008 [-0.004, +0.002]	+0.997 [+0.979, +1.015]

Table S20: Gradient-consistency diagnostics run when the defect was found.

Diagnostic	Question	Result
A1 forward shift invariance	$F(S + c\mathbf{1}) = F(S)$, softmax, LERP, Nearest, $K{=}2 \ldots 32$	Pass for all.
A2 gradient shift residual	$\rho = \lvert \sum_j g_{ij} \rvert / \sum_j \lvert g_{ij} \rvert$ per row	softmax $4.7{\times}10^{-8}$; LERP full $2.2{\times}10^{-7}$; LERP detach median 0.026–0.410, max near 1.
A3 all-ones directional finite difference (fp64)	autograd vs derivative along $s + c\mathbf{1}$	Detach $K{=}32$: autograd $-2.0{\times}10^{-3}$ vs true 0. Full: ≈ 0.
A4 interior finite difference	local LERP slopes away from extrema	softmax, detach and full all match.
P1-0 projection sanity	removes exactly the common-mode gradient?	Forward bitwise equal to detach; projected $\rho_{\max}$ $1.7{\times}10^{-16}$; $\max \lvert g_{\text{proj}} - g_{\text{full}} \rvert = 1.07{\times}10^{-3}$.

Order of events. The detached-gradient runs came first (5M-token sweeps, Table S22, then 100M runs with delayed divergence); diagnostics A1–A4 identified the detached extrema as an incomplete Jacobian, a `range_grad` switch with an unchanged forward was added, full-gradient reruns from scratch were stable and became the seed-1337 members of the five-seed matrix, and the P1 projection ablation followed. Every formal K-interval run uses the full calibration gradient (`range_grad=full` in every checkpoint config); the softmax runs use softmax's own backward, and the legacy `range_grad` field in their configs is never read by the softmax code path.

Table S21: Final-checkpoint geometry (layer mean, 8 validation blocks), detached vs full.

Condition	Backward	Span	h	$\|q\|$	Entropy	Top-level mass	Eff. levels	TV
Nearest K2	detach	226.2	113.1	58.0	3.97	0.679	1.81	0.439
Nearest K4	full	19.8	4.94	10.9	3.77	0.637	2.31	0.355
Nearest K4	detach	903.8	225.9	88.4	0.66	0.990	1.06	0.873
Nearest K8	full	25.5	3.18	11.3	3.86	0.506	3.57	0.259
Nearest K8	detach	1544.7	193.1	72.8	1.03	0.927	1.47	0.702
Nearest K16	full	29.6	1.85	12.2	3.78	0.386	6.70	0.160
Nearest K16	detach	5833.6	364.6	169.6	0.47	0.973	1.16	0.710
LERP K4	full	13.1	3.28	11.4	3.53	0.685	2.24	0.190
LERP K4	detach	69.0	17.3	30.6	1.87	0.903	1.44	0.513
LERP K32	full	21.9	0.69	12.0	3.72	0.334	13.15	0.017
LERP K32	detach	1192.6	37.3	130.1	1.12	0.825	2.35	0.281
LERP K32	detach + proj. (P1)	1632.2	51.0	78.4	0.22	0.979	1.17	0.278
softmax	native	25.1	—	12.2	3.79	—	—	—

5M confirmatory endpoints. Table S22 reports the six 5M-token Nearest runs with the detached backward against the same-seed 5M softmax reference (paired per block, same block bootstrap as the 100M endpoints). All six gaps are below 0.003 nats: at this horizon the endpoints had not yet exposed the later failure, which is not equivalence (five of six intervals exclude zero; cf. the short sweeps of Hernández-Cano et al., 2025), and these seeds are not the seed-1337 trajectory; the delayed onset is carried by the seed-1337 100M runs and its cross-seed replication by the LERP $K{=}32$ seed-7/42 runs.

Table S22: Historical 5M-token confirmatory runs, Nearest K with the detached backward, seeds 1338 and 1339 (124M, WikiText-103), against the same-seed 5M softmax reference: endpoint NLL over 243 validation blocks at 5.0M tokens, paired per block, 95% circular block-bootstrap CI (evaluation sampling only). Data: file `early_5m_seeds_detach.csv` in `E_detach_history/`.

Seed	K	NLL	softmax	Δ [95% CI]
1338	K2	6.9084	6.9055	+0.0029 [+0.0024, +0.0034]
1338	K4	6.9082	6.9055	+0.0027 [+0.0025, +0.0029]
1338	K8	6.9065	6.9055	+0.0010 [+0.0009, +0.0011]
1339	K2	6.9263	6.9236	+0.0027 [+0.0020, +0.0034]
1339	K4	6.9242	6.9236	+0.0006 [+0.0005, +0.0008]
1339	K8	6.9236	6.9236	+0.0000 [-0.0003, +0.0004]

Table S23: Detach replication, LERP $K{=}32$ at 124M @ 100M (WikiText-103): detach minus the matched same-seed full-gradient run at five checkpoints, and detach minus softmax at 100M; paired per block, 95% circular block-bootstrap CIs (evaluation sampling only).

Ckpt	seed 7	seed 42
20M	-0.002 [-0.006, +0.001]	-0.015 [-0.018, -0.013]
40M	+0.118 [+0.113, +0.123]	+0.074 [+0.069, +0.079]
60M	+0.672 [+0.653, +0.692]	+0.587 [+0.571, +0.604]
80M	+0.714 [+0.697, +0.731]	+0.781 [+0.764, +0.798]
100M	+0.802 [+0.785, +0.819]	+0.780 [+0.760, +0.799]
100M, detach − softmax	+0.807 [+0.789, +0.823]	+0.788 [+0.768, +0.807]

Replication on two further seeds. The headline condition, LERP $K{=}32$ with `-range_grad detach`, was retrained at 124M @ 100M on WikiText-103 with seeds 7 and 42 in the phaseB base code root (same code, data, schedule and protocol as the full-gradient five-seed members; detach runs with gradient checkpointing, their full controls without, step-0 check below), each paired against the existing same-seed full-gradient and softmax runs and evaluated through its own frozen code (243 blocks, batch 1). Table S23 gives the paired contrasts at the five evaluated checkpoints (circular block bootstrap, block 16, 4000 resamples); the detach and full trajectories of both seeds appear, against the full-gradient baseline, in Figure S9. Both seeds reproduce the endpoint (Table S23) and the shape: at 20M no degradation has appeared (seed 7 -0.002 [-0.006, $+0.001$]; seed 42 slightly *below* full, -0.015 [-0.018, -0.013], an interval that excludes zero), positive degradation is present by 40M ($+0.118$ and $+0.074$) and most of the gap is open by 60M; the final logged gradient norms are 2.4 and 2.3 against about 0.5 for the full-gradient runs. The legacy seed-1337 value ($+0.737$) comes from the older code root and is consistent with the two new seeds but is not a same-code replicate; a three-seed pool mixes lineages. The other detached conditions of the record remain single-seed.

Single-channel ablation (supplement, two seeds). With `range_grad` set to `max_only` or `min_only`, only one extremum's gradient is retained in the implemented backward; the forward

is unchanged. Four runs (LERP $K{=}32$, 124M, WikiText-103, 100M tokens; seeds 7 and 42) were trained in a copy of the phaseB base root on the same RTX 3090 / torch 2.5.1 as their same-seed full and detach controls and compared per validation block on that GPU (243 blocks, circular moving-block bootstrap, block 16, 4000 resamples; Table S25, Figure S9); CIs cover evaluation sampling only, and the two seeds are a replication, not a seed distribution, never merged into the five-seed tables. The detach controls were trained with gradient checkpointing, the full controls and new runs without; step-0 gradients were verified bitwise identical with and without it on the checked inputs, but the historical runs are not otherwise asserted to share every switch. The old and new full/detach implementations were regression-tested when the single-channel modes were added (bit-identical on CPU; GPU differences at the level of repeated executions). Per-step clipping fractions were not logged.

The two channels are not symmetric (Table S25): the maximum-only run recovers almost the whole detach–full gap on both seeds (0.98–0.99), ending within 0.015 nats of full with a softmax-like span and query norm, whereas the minimum-only run recovers almost none (0.003 and 0.072) and ends with a severely expanded span (layer means in the hundreds to thousands, like detach) and a gradient-norm rise; against detach it shows no detected endpoint difference at seed 7 and a small one at seed 42, and its larger span at a slightly lower loss shows that span and loss are not strictly monotone. The maximum-only gap against full grows slowly from 40M to 100M at seed 7 but not at seed 42, so no drift is claimed across seeds. Writing $w_j = f(z_j, h)$ with $z_j = s_j - M$ and $h = (M - m)/K$ on a locally smooth region, and treating the key's own score and the calibration variables as independent inputs,

$$\frac{\partial w_j}{\partial M} = -\frac{\partial f}{\partial z_j} + \frac{1}{K}\frac{\partial f}{\partial h}, \qquad \frac{\partial w_j}{\partial m} = -\frac{1}{K}\frac{\partial f}{\partial h},$$

so the maximum channel carries both the score-origin and the grid-width sensitivity while the minimum channel carries only the width term; retaining the maximum gradient therefore does not isolate a pure shift-anchor mechanism (index switches and extremum selection are handled by the implementation; these local expressions are not claimed at every boundary). Zero-sum residuals of the *implemented* score gradient (median over rows, mean over layers; for detach, minimum-only and maximum-only the modified backward, not the true derivative of the forward) at 100M are $5.5{\times}10^{-9}$ (full), 0.250 / 0.277 (detach, seeds 7 / 42), 0.257 / 0.233 (minimum-only) and $1.8{\times}10^{-3}$ on both seeds for maximum-only (about 65% of rows above 10^{-3}): strict per-row zero-sum is not necessary for near-full performance here, although the maximum-only residual is still two orders below detach, so an approximate zero-sum may matter; with the P1 result the zero-sum constraint alone does not explain the rescue, and shift invariance is not thereby shown to be irrelevant. The mean raw row minimum at seed 7 is -19.7 (full), $+2484$ (detach), $+1700$ (minimum-only) and -27.1 (maximum-only): a common shift of the raw scores is distinct from a change of the relative tail $m - M$. Two seeds of one condition identify the dominant channel in this setting, not a complete mechanism or a general necessity of the maximum channel.

Local gradient decomposition (supplement, diagnostic only). *Definition.* The 20M and 40M checkpoints of the two full-gradient LERP $K{=}32$ runs (seeds 7 and 42) were loaded without any optimizer step and the loss over the first 8 WikiText-103 validation blocks was back-propagated block by block ($\mathrm{loss}_b/8$, gradients summed). In the fixed-upstream analysis the upstream $u = \partial L/\partial P$ of every layer is recorded from the full model, and the local vector–Jacobian product of that layer's operator ($P = w/\sum w$, $w = \mathrm{reconstruct}(s, \mathrm{mode})$) is evaluated for each mode at the same scores and the same u. With g_{self} the implemented local VJP of the detach mode and g_M, g_m the contributions of the two extremum channels, $g_{\mathrm{full}} = g_{\mathrm{self}} + g_M + g_m$, $g_{\mathtt{max_only}} = g_{\mathrm{self}} + g_M$ and $g_{\mathtt{min_only}} = g_{\mathrm{self}} + g_m$ by construction (identity residual 0 at every layer; the M and m terms fall entirely on the arg-max and arg-min keys). *Implementation.* Boundaries, ties and the clamp follow the implementation. This local VJP is distinct from the whole-model in-graph backward, where a changed gradient in a later layer alters the upstream of earlier layers, so whole-model differences are not additive by the local identity. The diagnostic used deterministic algorithms (fp32 attention, bf16 autocast elsewhere, TF32 off); the formal runs did not. Reconstructing the historical P1 hook in-graph reproduced the historical P1 code bit for bit on these checkpoints and inputs, which validates the diagnostic implementation; no new P1 training run was made. *Result.* Table S24 reports norm ratios and relative errors $\|g_{\mathrm{mode}} - g_{\mathrm{full}}\|/\|g_{\mathrm{full}}\|$ per layer, averaged over the 12 layers. The maximum-dependent term dominates the omitted calibration correction (norm ratio 0.36–0.52

against 0.008–0.009 for the minimum-dependent term; norms, not additive shares); preserving it leaves a layer-averaged local relative gradient error below 1%, consistent with the near-full training performance of `max_only`, while the minimum-only error is indistinguishable from detach. The detach error is concentrated on the arg-max key (about 99.9–100% of its squared error, rounded), which describes where detach departs from full, not the full gradient itself; a row-zero-sum projection applied locally ("P1 (local)") leaves most of this local discrepancy intact and is not the historical in-graph hook. *Limits.* Not a per-row or per-layer guarantee and not a bound on whole-model score-gradient error; at 20M no clear loss degradation has appeared while at 40M an early separation is present (detach − full $+0.118$ at seed 7, $+0.074$ at seed 42), and reading checkpoints of the full trajectory does not establish a temporal order along the failing trajectories; parameter-gradient norm differences are not translated into update differences.

Table S24: Fixed-upstream decomposition of the score gradient at the 20M and 40M checkpoints of the full-gradient LERP $K{=}32$ runs (seeds 7 and 42; first 8 WikiText-103 validation blocks; deterministic diagnostic process). Left: norm of each term relative to $\|g_{\text{full}}\|$, per layer then averaged over 12 layers (norm ratios, not additive shares). Right: relative error $\|g_{\text{mode}} - g_{\text{full}}\|/\|g_{\text{full}}\|$ of the implemented local gradient, per layer then averaged; "P1 (local)" is a post-hoc row-zero-sum projection of the local detach gradient, not the historical in-graph hook. Data: `grad_decomp_local_layers.csv` in `round2/task2` and `round2/task2_s42` of `supplement_four_runs_20260923`.

		norm ratio to g_{full}			relative error of the implemented gradient			
Seed	ckpt	self	M term	m term	detach	min_only	max_only	P1 (local)
7	20M	0.909	0.361	0.0083	0.361	0.361	0.0083	0.353
	40M	0.807	0.509	0.0092	0.509	0.509	0.0092	0.507
42	20M	0.897	0.384	0.0079	0.384	0.384	0.0079	0.376
	40M	0.792	0.522	0.0092	0.522	0.522	0.0092	0.520

Table S25: Single-channel ablation, LERP $K{=}32$ (124M, WikiText-103, 100M tokens; seeds 7 and 42; every run and its same-seed controls evaluated on one RTX 3090). ΔNLL in nats against the same-seed control, paired per validation block, 95% circular moving-block bootstrap CIs (block 16, 4000 resamples) over 243 blocks: evaluation sampling only, not seed-to-seed variation. Span: layer-mean row span at 100M over 8 blocks. Gap recovery per seed $= (\text{NLL}_{\text{detach}} - \text{NLL}_x)/(\text{NLL}_{\text{detach}} - \text{NLL}_{\text{full}})$. Supplementary runs, not merged into the five-seed tables.

Seed	Mode	NLL @100M	− full	− detach	span	gap recovery
7	full (control)	4.1415	—	—	21.4	1.000
	detach (control)	4.9437	+0.8021 [+0.7849, +0.8202]	—	1055.6	0.000
	min_only	4.9414	+0.7998 [+0.7817, +0.8188]	-0.0023 [-0.0074, +0.0031]	1488.8	0.003
	max_only	4.1566	+0.0151 [+0.0127, +0.0174]	-0.7870 [-0.8047, -0.7703]	26.5	0.981
42	full (control)	4.1511	—	—	21.5	1.000
	detach (control)	4.9311	+0.7799 [+0.7605, +0.7993]	—	860.7	0.000
	min_only	4.8751	+0.7240 [+0.7095, +0.7388]	-0.0559 [-0.0613, -0.0504]	1738.6	0.072
	max_only	4.1602	+0.0091 [+0.0070, +0.0114]	-0.7708 [-0.7889, -0.7523]	25.9	0.988

G ASSETS, CODE AND REPRODUCTION

Runs added after the formal suites (outside the formal total). Fifteen complete training runs were added after the formal suites, all 124M @ 100M on WikiText-103 with the suite-D schedule and five checkpoints each (75 checkpoints): the two detach-replication runs (seeds 7 and 42, Appendix F), one exploratory WindowFloor–Weight $K{=}4$ run (seed 1337; completed and evaluated, then replaced by the tail-policy decomposition because it changed two factors at once) and the twelve supplementary runs reported in this paper (60 checkpoints): four single-channel runs, seeds 7 and 42 (Appendix F; RTX 3090; the seed-42 pair registered before training, an aborted first attempt without any checkpoint recorded only in the execution log), four strict-forward runs (Appendix C.2; RTX 5090), two tail-policy runs (Appendix D.4; RTX 3090) and two reduced-precision exponential baselines (Appendix D.5; RTX 3090; registered before training, both ending at step 763, 100,007,936 tokens; the native-FP8 backward check is a diagnostic, not a training run). The frozen-endpoint audit and the gradient diagnostics re-evaluate existing checkpoints and add no training. Manifests, registration addenda, code roots, scripts and result files of every item are listed with full paths in `paper_training_v1/compact/REPRODUCTION_INDEX.md`, with the local asset audit (`ASSET_ANALYSIS_AUDIT_20260925.md`, `manifests/local_training_asset_a`

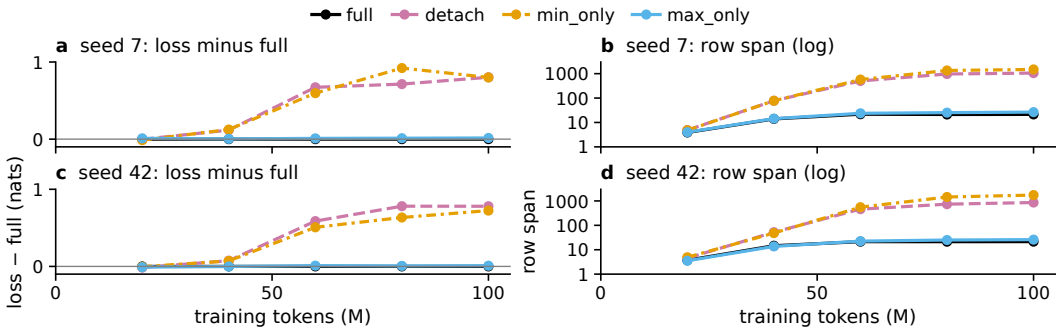


Figure S9: **Single-channel ablation trajectories**, LERP K=32, seeds 7 (a, b) and 42 (c, d): new single-channel runs paired with the existing same-seed full and detach controls on one GPU (the seed-1337 record of Figure 2 is a different lineage). (a, c) Validation loss minus the same-seed full-gradient run at the five evaluated checkpoints; (b, d) layer-mean row span (log). The baseline is the same-seed full-gradient run, not softmax; the detach − full values at these checkpoints are tabulated in Table S23. All series are read from the per-checkpoint evaluation files.

Table S26: Formal assets and later additions, counted in complete training runs and in evaluated checkpoints. Every formal K-interval run uses the full calibration gradient (`range_grad=full` in its checkpoint config); the softmax runs use softmax's own backward, and the legacy `range_grad` field in their configs is not read by the softmax code path (Appendix F).

Asset	Runs	Checkpoints	Status
A: 124M @ 2.5B, FineWebEdu-3B, seed 1337	15	300	formal; downstream battery
B: 124M @ 250M, FineWebEdu-3B, seed 1337	10	100	formal
C: 1B @ 100M, WikiText-103, seed 1337	12	60	formal
D: 124M @ 100M, WikiText-103, seeds 7/42/253/1337/2026	130	646	formal
Formal total	167	1,106	all evaluated, 0 failures
Historical detach intervention (Appendix F)	16	28 eval. records	separate record, not in the formal total
Superseded (five-seed cells trained on FineWebEdu-3B)	40	200	replaced by the WikiText-103 rerun; 240 checkpoint paths, of which every `ckpt.pt` is a hard link to the 100M step checkpoint
Supplementary runs used in this paper (below)	12	60	outside the formal total
Detach replication, LERP K=32 seeds 7/42 (Appendix F)	2	10	outside the formal total
Superseded exploration: WindowFloor–Weight K=4, seed 1337	1	5	completed and evaluated; replaced by the tail-policy decomposition, not used as a mechanism control

`udit_20260925.tsv`). The supplementary runs are reported separately and never merged into the five-seed tables.

Verification counts. As recorded by the gate and audit reports: the suite-D integrity gate checked 130 runs and 646 evaluation jobs with 0 errors; suites A–C comprise 37 runs and 460 checkpoints, all evaluated with 0 failures; the downstream battery comprises 105 harness units and 1,399,470 per-item rows with 0 failures; the 2026-09-25 local audit lists 243 run directories (224 complete trainings, 940 step checkpoints), records no missing evaluation for any run that was due one, and confirms that suites A–C and E2 were evaluated from the cloud originals.

Historical labels and frozen code roots. Result files, run names, directory names and code fields keep their original identifiers. Calibration: `QRM`, `FW` and `range_mode=relevant_zero` are legacy identifiers of the zero-tail window mode and display as FWM; reconstruction: `nearest` displays as Nearest (nearest-center rounding); in legacy run names `_fullgrad` denotes the full calibration gradient and does not by itself identify the STE placement. The surrogate identity lives in `quant_attention.py`, each run is evaluated through the code root it was trained with, and the labels Weight-STE and Prob-STE are assigned only when the saved configuration and the SHA-256 prefix of that file establish the surrogate: `9d03d7a6` (softmax, LERP, MinMax–Weight) and `a402460d` (FWM–Weight) implement the weight-level replacement `w = w_lerp + (w_nearest - w_lerp).detach()` → Weight-STE (result-file labels `MinMax-Weight`, `QRM-Weight`); `0e20aea0` (MinMax–Prob) and `ce2bd02b` (FWM–Prob) implement the probability-level replacement (run suffix `_probste`) → Prob-STE; `3da199f8`

is the legacy seed-1337 softmax. The hashes are authoritative when a legacy directory name is ambiguous; paths, hashes and checkpoint config fields are not renamed. The formal code roots of suites A–C (one per surrogate/calibration family, plus one that only enables $K{=}1$ for LERP) have a MinMax path hash-identical to the full-gradient implementation of the five-seed suite.

Manifests and reproduction. The suite manifests (`manifests/primary_37_runs.tsv`, `primary_37_checkpoints.tsv`: 37 runs, 460 checkpoints; `D_.../manifests/five_seed_wikitext_130.tsv`, `five_seed_checkpoints_130.tsv`: 130 runs, 646 jobs) carry operator axes, seed, code root and checkpoint path; everything downstream reads identity from these, never from a directory name. `excluded_runs_130.tsv` is the frozen gate-time list of the 43 excluded runs with reasons (40 FineWebEdu-3B originals superseded by the WikiText-103 rerun, one invalid-environment copy, one archived backup, and the P1 projection run, which is not a full-gradient run); the 2026-09-25 asset audit found one further directory absent from that list, an aborted 20M copy in the same invalid environment that never entered any manifest, gate or analysis, recorded in the dated addendum `excluded_runs_addendum_20260925.tsv` (the frozen list is unchanged). Counts distinguish complete training runs, checkpoint files, evaluation jobs, repeated evaluations of one checkpoint on a second GPU, and model-only weight copies; only the first two enter the totals above. Evaluation, analysis, table and figure commands, with the frozen-code and offline-harness requirements: `paper_training_v1/compact/REPRODUCTION_INDEX.md` (§4).

H Extended related work

Approximate softmax operators. Softermax (Stevens et al., 2021) and I-BERT (Kim et al., 2021) replace the exponential with a base-2 piecewise or integer-polynomial approximation and fine-tune downstream; their scales are static and never receive a gradient. ConSmax (Liu et al., 2024) trains a 6-layer model from scratch with learned per-head parameters in place of the row reduction, keeps the exact exponential (so its calibration quantities are optimizer-owned) and attributes its own early instability to non-unit normalization. Two base-2 softmax papers (Zhang et al., 2022; 2023) study the classifier softmax: the first argues trainability from gradient structure (a base change scales the cross-entropy Jacobian by $\ln 2$); the second finds that i.i.d. error above 10^{-6} injected into an exact-backward pipeline breaks training, whereas our forward error is a deterministic function of the scores. Our operators keep $\sum_j P_j = 1$ in exact arithmetic by normalizing after reconstruction, a first-order correctness criterion in Zhang et al. (2022). IndexSoftmax (Zhong et al., 2026) replaces the exponential with a fixed-window 32-entry lookup table and normalizes in integers, without retraining.

Straight-through estimators and calibration gradients. The straight-through estimator was named and analysed by Bengio et al. (2013) for stochastic binary neurons; binarized networks (Hubara et al., 2016) made it the standard training rule for hard forwards; Yin et al. (2019) analyse a two-layer model with binary activations. The temperature of straight-through Gumbel-softmax (Jang et al., 2017; Maddison et al., 2017) is the closest analogue of our h, except that it is annealed by the practitioner whereas h under MinMax is set by the model's own score geometry. TQT shows that the TensorFlow fake-quant path zeroes the threshold gradient inside the interval, so the bounds only move outward (Jain et al., 2020); the PyTorch fake-quantization path computes its observer statistics on a detached tensor (PyTorch contributors, 2026). ITA (Islamoglu et al., 2023), the closest attention-side instance, computes a base-2 softmax on 8-bit integers by shifts, observes that above a certain input scale all entries but the row maximum quantize to zero, and tunes the clipping range by quantization-aware training with that softmax in the loop; it learns a quantizer scale, not a per-row quantity inside the operator, and does not analyse the backward. These observers are running buffers, so the analogy is structural; what it locates is the graph choice we test: whether the derivative of a per-row extremum used inside the forward is propagated back to the scores that produced it.