ONE QK CHANNEL, MANY SOURCES: TRACING LOW-PRECISION ATTENTION COLLAPSE

Anonymous authors
Paper under double-blind review

ABSTRACT

A bfloat16 transformer can train normally for many steps, then collapse abruptly. Prior work links the collapse to attention errors whose shared structure lets rounding contributions compound, and shows QK normalization disrupts that compounding. Distinct low-precision errors trigger the same collapse, leaving unclear whether each needs a fix at its source or one shared route can be blocked instead. We isolated the fault behind a reproduced GPT-2-class collapse to the streaming-softmax accumulator, where an fp32 streaming core repairs it, and turned it into an assay for moving a controlled error across sources. Using it, we found that errors placed outside attention still drove the same QK spectral runaway, and that correcting only QK kept training stable while the fault stayed active. This is a *source-channel dissociation*: fault source is not failure channel. It held across tested architectures and scales, and reproduced on a second GPU architecture. As a causal probe, projecting each update off the current QK weights' leading three singular directions held the query projection's largest singular value to 11.1, whereas removing equal energy elsewhere left it at 237: the QK channel causally drives the early runaway. What lets the injected error in is *temporal sign-coherence*, its per-head sign persisting across steps, not aggregate deviation; once inside, the runaway shows as attention-logit saturation. If one channel carries these faults, a QK-local action switched on mid-run should contain each. *QK-Guard*, a dormant controller, tests this by switching on parameter-free QK normalization at the first monitored threshold crossing. On the runs designated for this test, the QK-local action prevented the failure of each matched or same-configuration unguarded run; on plain GPT-2, all 12 final train and validation losses were within 0.03 nat of same-configuration always-on QK-norm, and both methods ran 60k steps without collapse. Intervention at the QK locus therefore suffices in place of a fix at each source.

1 INTRODUCTION

Bfloat16 transformer training is routine (DeepSeek-AI, 2024), yet a run can appear healthy through long optimization before abruptly collapsing, revealing little about what accumulated. Prior work links this pattern to structurally aligned attention errors with compounding rounding contributions and identifies query–key (QK) normalization as what disrupts that compounding (Qiu & Yao, 2026). The same normalization also prevents attention-logit growth, a known instability of large-scale training (Dehghani et al., 2023; Wortsman et al., 2023).

The open question is whether low-precision faults require repair at their sources or instead share a blockable failure channel; collapse alone cannot decide, since it reveals only the endpoint. We therefore pair controlled fault-source moves with always-on module-routed gradient correction in an identification assay that varies the fault source independently of the route receiving a clean fp32 gradient, with the rest fixed and each source mechanism left enabled, so the splice tests channel blocking while every source stays unrepaired. Faults within and outside attention converge on the same QK spectral runaway, and correcting only the QK route blocks that channel. This source-channel dissociation is our thesis: fault source is not failure channel.

Together, these results change what counts as a sufficient remedy: because heterogeneous sources share the QK channel, one action at the QK locus can replace a separate repair at each source. An

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.

fp32 streaming core fixes the streaming-softmax accumulator fault but not post-reduction weight-gradient quantization or errors outside attention. We test that implication in flight with QK-Guard, which watches the downstream attention-logit saturation because ordinary feature learning is temporally coherent too.

Our contributions are three findings about the QK channel, each turning on a discriminating contrast.

1. **A source-channel dissociation.** We pair controlled fault-source moves with always-on module-routed gradient correction while leaving each source mechanism enabled: faults originating within and outside attention converge on the same dominant QK channel, which correction on the QK route blocks and correction on the MLP route does not.
2. **The entry condition for the QK channel.** We manipulate error-sign sequences to identify temporal sign-coherence (the persistence of each head's error sign across optimizer steps) as what decides whether an injected error enters the QK channel, separating it from the per-tile scalar mean and aggregate magnitude that prior analysis leaves coupled to it; against an equal-energy off-target control, removing each run's own dominant QK weight directions verifies that the channel causally drives the early runaway.
3. **Containing the channel in flight.** QK-Guard, a dormant closed loop, tests whether acting at the QK locus mid-run contains these faults: it watches attention-logit saturation and, when its monitored statistic first crosses the threshold, applies parameter-free per-head QK normalization. Across the runs designated for this test it prevents the failure of each matched or same-configuration unguarded run without repairing the fault or rewinding the run, and at the same trigger a softmax-local dynamic maximum and a learning-rate cut fail, so reactivity alone does not determine rescue.

2 Isolating a Fault and Building the Assay

To edit one numerical factor at a time, we reproduced the GPT-2-class collapse reported by Qiu & Yao (2026) on GPT-2 small (Radford et al., 2019) trained on OpenWebText (Gokaslan & Cohen, 2019), in a PyTorch re-implementation of FlashAttention-2's tiled online-softmax recurrence (Dao, 2023) derived from that report's public code. Its carries are the running output O and softmax statistics (stats), the running maximum and exponential sum (Milakov & Gimelshein, 2018). Vendor kernels keep these carries in fp32; this implementation stores them in bf16 and exposes the recurrence for controlled edits, which lets a fault stay active while one route at a time is cleaned. Table 1 groups the edits by the uncertainty removed. The kernel's maintainer had publicly flagged accumulator precision as one untested implementation difference (Dao, 2025). The streaming core's precision is central: three protocols observe no collapse once the same tiled implementation runs its core in fp32: that arm at two seeds, the fp32/fp32 carry condition at two seeds, and one long run manually stopped while healthy at 103400 logged steps. The fp32-core intervention keeps tiling on while changing the outcome, so tiling is not the factor; the full-matrix arm corroborates. Together with panel (b), where either bf16 carry alone suffices to collapse, these contrasts localize collapse to a bf16 streaming-accumulator artifact rather than bf16 operands, the FlashAttention kernel family, tiling, or a single-scale artifact.

The two carries enter backward differently. The output carry reaches dQ and dK, never dV: saved O re-enters only through the D-term $D_i = dO_i^\top O_i$, the per-row inner product of the output gradient with O, which enters the score gradient as $dS_{ij} = P_{ij}(dP_{ij} - D_i)$ (Dao et al., 2022) (four-bit attention training keeps a high-precision output for this term alone (Zhang et al., 2026)). The statistics carry reaches dQ, dK, and dV through the saved log normalizer. Either carry produces the same $\sigma(W^Q)$ runaway; the split decomposes one streaming-accumulator source with its generator fixed and leaves open whether other sources converge on it.

A manipulable assay. Only the output carry yields an exactly computable injection: its bf16-minus-fp32 saved-output difference enters through the per-row D-term while an fp32-core forward pass keeps the residual stream clean, and it is recomputed each step from the current Q, K, and V, so the error tracks current weights and data rather than a stored tensor. That makes the error movable along four axes: source, placement (an emulation sub-choice within a source), dose, and structure. Scaling between the fp32-core and bf16-accumulation endpoints turns the same-code contrast into a dose continuum.

Table 1: Localizing the collapse. Each row changes the factor it names; the streaming-core row keeps tiling on while the core's precision changes. "Coll." is collapsed over tested runs under the sustained-divergence rule (Appendix A); × collapse, ▶ a last observed step without collapse (an observation bound). SDPA is PyTorch's fused path. In panel (b), both collapsing carries show the same runaway in $\sigma(W^Q)$, the query projection's largest singular value.

(a) Localizing the fault

CONDITION	COLL.	STEP
The object, and the fp32-core intervention on it		
bf16 O + bf16 stats	6/6	× 700–1400
streaming core → fp32	0/2	▶ 4900
Scope: is another factor sufficient?		
SDPA, bf16 operands	0/2	▶ 4900
FlashAttention-2 forced	0/2	▶ 4900
Full matrix, no tiling	0/2	▶ 4900
Scale: GPT-2 medium		
re-implementation	2/2	× 900, 1400
SDPA	0/2	▶ 4900

(b) Which streaming carry: three observed conditions

O / STATS DTYPE	COLL.	STEP
bf16 / fp32	2/2	× 700, 1300
fp32 / bf16	2/2	× 2300, 2600
fp32 / fp32	0/2	▶ 7900

Under backward-only placement, reduced doses delayed collapse monotonically to 2800, 5300, 6800, and 10100, in that order (Appendix D). Over a pre-registered window of healthy-phase states and eight batches, the median one-step relative ℓ_2 error in dW^Q against an fp32 reference was 0.734 under the bf16-accumulation re-implementation and 0.0164 under fp32-accumulating SDPA. In one seed, a matched-recipe fp32-core arm was stopped while still healthy at 103400 steps, whereas its untreated control collapsed at 700. An fp32 streaming core is therefore effectively immune.

The assay supports two tests: §3 replaces bf16 accumulation with post-reduction weight-gradient quantization across module targets inside and outside attention, everything else fixed, after first locating where the fault must be intercepted; §4 applies matched-magnitude structure transforms with everything else fixed.

3 IDENTIFYING A COMMON QK FAILURE CHANNEL ACROSS FAULT SOURCES

To test whether repair must act where error originates, we invert two manipulations. Route correction keeps §2's accumulator fault active while moving the repair through a clean-gradient splice; source shift fixes the repair while replacing bf16 accumulation with a different error at a different site; together they test whether heterogeneous generators converge on one query–key channel that one action blocks while each generator remains active.

3.1 LOCATING THE BLOCKING CHANNEL

Each step runs the same weights and batch backward through the faulty re-implementation and the fp32-core reference, splicing the clean gradient into one named route while leaving all others faulty. Each row therefore tests whether its named route must be intercepted; Table 2 shows that the anticipated architecture-specific rescue does not occur.

Cleaning only the query–key route rescues every tested architecture—plain GPT-2, a RoPE variant (Su et al., 2021), a 350M model, and a LLaMA-style architecture (Touvron et al., 2023)—to train losses from 3.23 to 3.88; MLP cleaning rescues none, and on LLaMA whole-attention cleaning reaches the same train loss as the query–key route (3.23 in both), placing the effective position inside that block. A four-seed LLaMA check without splicing gives 4 of four untreated collapses

and 0 of four under per-head query–key normalization (Henry et al., 2020). Moving the fault to the kernel's other carry preserves the split: bf16 softmax statistics, the path that also reaches dV, collapse two untreated route-grid seeds at 2600 and 2400; query–key cleaning keeps both healthy through 7200 and 11900 logged steps, whereas MLP cleaning collapses at 3000 and 1900. Thus query–key correction is the route-level blocking intervention across the tested grid.

3.2 Moving the fault source

Because every prior arm shares one generator, QK's blocking role could be specific to bf16 accumulation, so we fix the intervention and change the generator: the second one rounds the materialized weight gradient after backward and reduction, before clipping and the optimizer step, onto an MXFP4-style grid (Rouhani et al., 2023) of four-bit elements in blocks of thirty-two sharing a power-of-two scale. Attention itself runs unchanged under bf16 autocast through the fused SDPA path (bf16 operands, fp32 accumulation) in every arm, so the selected module gradient is the only injected low-precision source; attention's own rounding error is present in all arms, including the QK-normalized twins.

Applied to the query–key gradient alone, the quantizer drives the same spectral runaway in W^Q, to 292 and 205, with attention logits reaching 1.73e+06; per-head query–key normalization suppresses it to 24 and 26 at healthy validation losses of 3.39 and 3.41. At a matched grid and magnitude, deterministic and stochastic rounding (Gupta et al., 2015) both produce the runaway signature; loss failure within this shorter window is seed-variable, and the separately powered all-weight-gradient extension through 20k records 11 of 12 quantized collapses, versus 0 of 4 with query–key normalization.

Moving the identical emulator, block size, and per-step seed schedule to MLP and value gradients yields 4 of four unprotected collapses on two data-parallel ranks after the same W^Q runaway (Figure 1a; other rank counts in Appendix D). Per-head query–key normalization leaves 0 of four collapsing, holds the runaway near 30, and brings both targets to a healthy 3.24 validation loss. Changing only the target, the matched sweep shows two generators reach one channel; one action contains both. The weight-gradient quantizer spans four sites, three outside the query–key path.

Applying the same four-bit grid to the GEMM operands, with fp32 accumulation, leaves training healthy through 11900 steps with the spectral norm no higher than 39: four-bit arithmetic is not by itself the trigger in these configurations, an observed boundary at a milder dose (Appendix D).

3.3 Portability and scope

On a single Blackwell B200 running our exact training code, software-emulated quantizer included, the dissociation reproduces: an off-attention MLP source still develops the query–key runaway and collapses, while per-head query–key normalization triggered mid-run holds a second run of the same faulted configuration healthy (Figure 4; per-arm steps and the eager control in Appendix D).

RoPE, RMSNorm and scale delay this failure without preventing it. The delay depends on RoPE, but no combination of the three prevents the outcome; safety here comes from an fp32 attention core or from query–key normalization. Table 2(b) gives the ladder. The failure also appears with SGD in place of AdamW: the untreated fault collapses while the matched clean controls do not (Appendix A).

The dissociation these results establish has a falsifiable bound: the query–key channel is the dominant attractor for coherent low-precision error that perturbs residual-block parameter updates or enters upstream of those blocks, because in this pre-norm setting the query–key projections read a normalized input. The falsifier follows: a source acting after the final attention module has no later query–key computation to be intercepted at, and would escape. Within that scope every source we moved the fault to reached the same channel. That scope leaves the gate unresolved: whether coherence or sheer magnitude opens the channel.

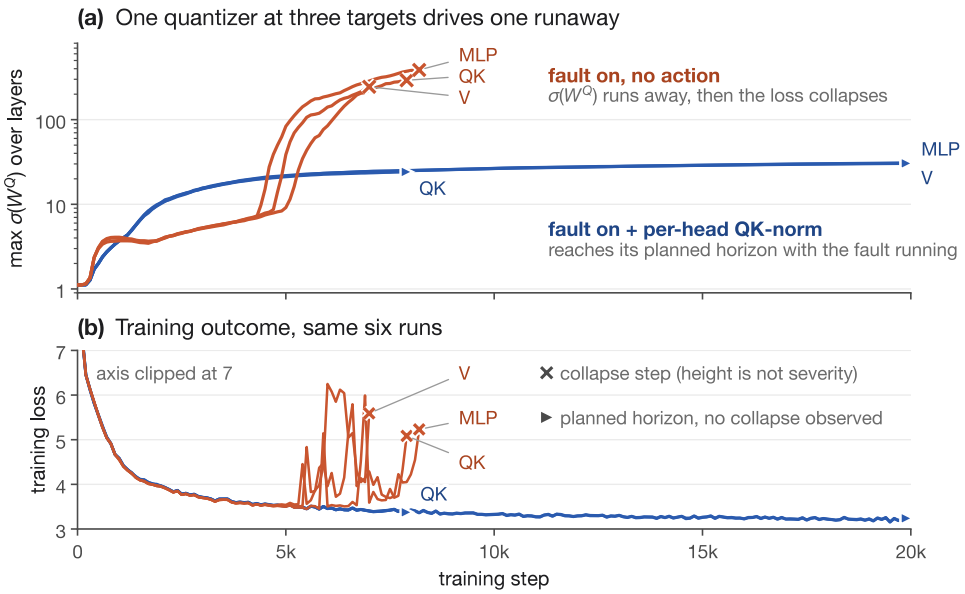

Figure 1: Source-channel dissociation. **(a)** one weight-gradient quantizer routed to the QK, MLP or value weight gradient drives the same $\sigma(W^Q)$ runaway (one quantizer at three locations); per-head QK-norm twins survive with the fault running, the action capping the attention logit while leaving σ uncapped (Figure 4). **(b)** outcomes on the same step axis. Labels name the target at each run's end; one seed per arm, planned horizons 7900 or 19900 steps; $\times$ a collapse step (height is not severity), ▶ a planned horizon.

Table 2: Route correction across architectures. Panel (a): collapsed over seeds run; empty cells not run. Panel (b): the same untreated fault per architecture, the last row adding per-head QK-norm to RoPE + RMSNorm. †The non-collapsing seed plateaus at 4.50 train loss through 12800 steps, above every QK-corrected final (3.23–3.88). ‡Seed-variable (one RoPE seed healthy, one plain-GPT-2 run truncated while diverging); no claim rests on this row.

(a) Which route must be corrected?

GRADIENT ROUTE CLEANED	PLAIN	+ROPE	350M	LLAMA
none (fault untreated)	1/1	2/2	2/2	1/1
QK **only**	**0/2**	**0/3**	**0/2**	**0/1**
MLP only	2/2	2/3†	2/2	1/1
V and O	1/1‡	1/2‡	—	1/1
whole attention block	—	0/1	—	0/1

(b) Architecture moves the timescale; QK-norm moves the outcome

CONFIGURATION	COLLAPSED, STEP
RMSNorm alone	1400
RoPE alone	7700
RoPE + RMSNorm	13300 and 13700
four further seeds	4 of four
minus RoPE	1400 and 1900
+ QK-norm	none of 4 seeds, to 19900

4 EXPLAINING ENTRY AND TESTING THE CAUSAL ROLE OF THE QK CHANNEL

§3 located where failure can be blocked; it left open what admits the large, structured errors observed there and whether the query–key channel's directions drive it. Structure interventions at matched

magnitude, where possible, answer the first question; removing the channel's dominant directions answers the second.

4.1 Temporal sign-coherence, not aggregate deviation, gates entry

We return to the injection assay of §2 at one magnitude and one width, applying one predefined structure transformation to the injected backward error per cell at matched per-injection magnitude. Left unaltered, the error collapses training in 5 of five seeds, between 1000 and 2900 steps. Scrambling its sign element by element at the same magnitude collapses 0 of four (two seeds censored at 7900 steps, two healthy at 19900, ending at 3.33 and 3.36). Permuting the error along the batch axis, which preserves the batch-summed tensor at every head and position while reassigning values across examples, collapses 3 of three, between 1000 and 1300 steps. Gaussian noise at matched magnitude collapses neither of two seeds through 11900 steps. Destroying which example a contribution came from does not prevent the collapse; destroying the sign does at the tested natural scale. Each alternative reading is excluded by a different arm: batch permutation excludes example identity, the matched-magnitude scramble and noise arms exclude magnitude, the mean-removal pair below excludes the scalar mean, and a smaller dose does not rescue the coherent error, which at half the magnitude still collapses on three seeds in the natural context, at 1600, 1800 and 2000 steps.

A second pair separates two quantities that a prior derivation of this collapse leaves coupled (Qiu & Yao, 2026): a mean component of the backward error and the error's temporal structure. Injecting only the error's scalar mean, reduced over each tile, is benign, holding two seeds to 9900 and 8600 steps at 3.31 and 3.40, while the coherent error with that mean removed still collapses both seeds, at 5100 and 9200 steps through the same $\sigma(W^Q)$ runaway, reaching 76.8. Neither aggregate magnitude nor that scalar gates entry. The derivation of Qiu & Yao (2026) reaches its conclusion through coherence and already names query–key normalization as the disruptor; what this pair adds, for our tile-reduced scalar, is that the two quantities can be moved independently (Appendix C).

The mean matters one level down: per-head DC bias, the same reduction with the head axis retained, is a sufficient carrier. Injected alone, reduced over the whole pass and across ranks in the natural low-precision forward, it collapses all 6 full-horizon seeds, from 2500 to 11800. Removing it protects two seeds through 9900 steps in the clean-forward surrogate (forward pass kept clean), while full-pass removal in the natural forward still collapses at 1100 and 1900 steps: these assays show necessity in the clean-forward surrogate and sufficiency in the natural forward, and a separate near-threshold per-tile per-head backward-only injection also collapses two of four seeds.

Sign randomization protects under two complementary constructions: whole-tensor randomization holds both arms to 8900 steps at 3.21 and 3.25, and per-head signs drawn once per optimizer step, which hold each head's sign fixed within the step and so keep its exact per-step carrier magnitude, hold both seed-matched arms to the same horizon. With the four matched sign-scramble survivors and three half-dose true-sign crashers, these rule out a magnitude-only explanation and jointly identify temporal sign persistence as a load-bearing tested discriminator in this regime (Table 8, Appendix C).

4.2 Direction-specific removal suppresses the QK runaway

Everything so far changes what the channel receives; whether the channel itself drives the runaway is still open. The last experiment removes the channel's own dominant directions: at each optimizer step we form the three leading singular directions of the current W^Q and W^K, project that step's realized update off them, and rescale the remainder to the update's original norm, so the update loses its component along the weight's leading directions at unchanged total energy. Against the weight-gradient sources of §3.2 this suppresses the runaway: all four treated arms hold $\sigma(W^Q)$ between 11.1 and 15.4 and none collapses through 15900 steps, ending between 3.25 and 3.26 validation loss.

Suppression alone would not settle direction, because removing any energy from the update might do it. The control removes equal energy from outside that subspace, leaving the target directions in place, at matched per-step update norm and intervention magnitude, and does not suppress (Figure 2): across the same four cells $\sigma(W^Q)$ runs to between 237 and 491, the two model-wide arms collapse at 8100 and 15600 steps, and the two query–key arms carry the runaway without collaps-

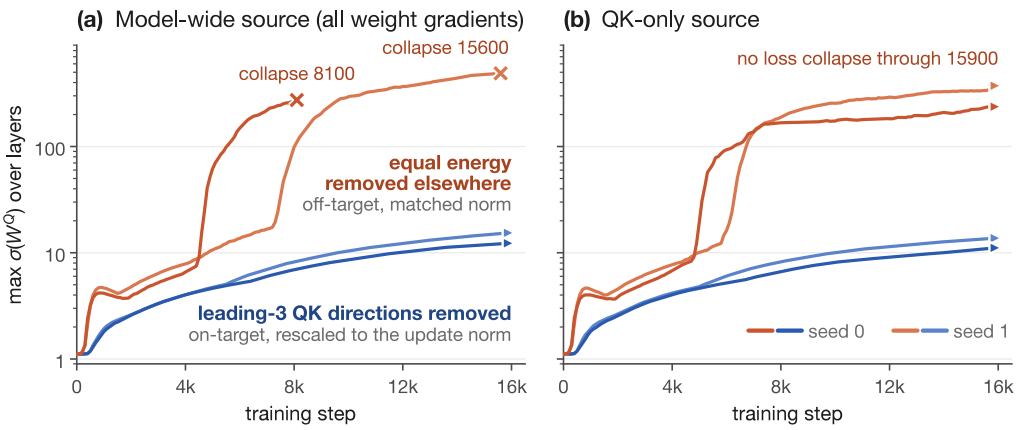

Figure 2: Direction-specific removal suppresses the QK runaway. Each step's realized update is projected off the three leading singular directions of the current W^Q and W^K and restored to its norm; the control removes equal energy from outside that subspace at matched norm. On-target removal suppresses the runaway under **(a)** a model-wide and **(b)** a QK-only source; the control does not. Two seeds per arm (seed 0 darker, seed 1 lighter shade); $\times$ marks a collapse step; in (b) the off-target arms run away without loss collapse inside this schedule (▶ at 15900 steps).

ing inside this schedule (0 of 2 through 15900 steps), with four fresh seeds of that cell on a longer schedule collapsing 4 of four. Merely shrinking the update without targeting a direction does not rescue either: all four such arms collapse, between 9000 and 11600 steps, so net reduction of update energy is neither necessary nor sufficient. The same specificity holds at 350M: on-target removal carries two seeds to 19900 steps at 19.0 and 22.0, while the equal-energy off-target pair collapses at 13000 and 16700 steps.

The query–key channel is therefore a causal driver of the early spectral runaway, and of the collapse that follows it inside these horizons. The removal serves as a causal probe; §5 supplies the remedy. The sources share one query–key-local channel whose directions move with the training state: their runaway directions overlap only partially (Appendix D).

5 Closing the Channel in Flight

5.1 Monitor downstream, act at QK

Acting at the QK locus in flight requires choosing when: cumulative QK update-coherence sits near unity in doomed and safe runs alike, with overlapping ranges and even inverted minima, because ordinary feature learning is temporally coherent too. The spectral norm the runaway acts on is no better as a trigger, since healthy and doomed $\sigma(W^Q)$ ranges also overlap and an acceleration rule misses the spectrally flat RoPE collapse (nor is high $\sigma(W^Q)$ sufficient for loss failure at deployment length, Appendix C). QK-Guard therefore monitors the runaway's downstream attention-logit signature, permanently switching on parameter-free per-head QK normalization at the QK locus at its first monitored crossing of the fixed threshold of 30. The statistic is, per layer, the mean over a fixed four-sequence validation probe of each query's largest causal logit, maximized over layers, read every 50 to 200 steps (per arm family) on the post-update weights, so a fire is the first periodic reading above 30 and normalization is active from the next step (Appendix A). The value was fixed once, on the plain GPT-2 fault, before the reported guarded transfer runs were launched, and never re-tuned; Appendix A gives its provenance and a two-seed sweep from 10 to 100 on the plain fault and the held-out family (every setting fires, and no guarded arm collapses through its last logged step while both untreated plain seeds collapse). On the reference arms it was set on, it separates converged healthy envelopes from doomed pre-collapse trajectories. It serves as a trigger: a fire switches on the normalization that always-on training carries from step zero (Appendix C reports a fire on fault-free training). On plain GPT-2, both unguarded seeds end at 10.61 and 9.86 train loss,

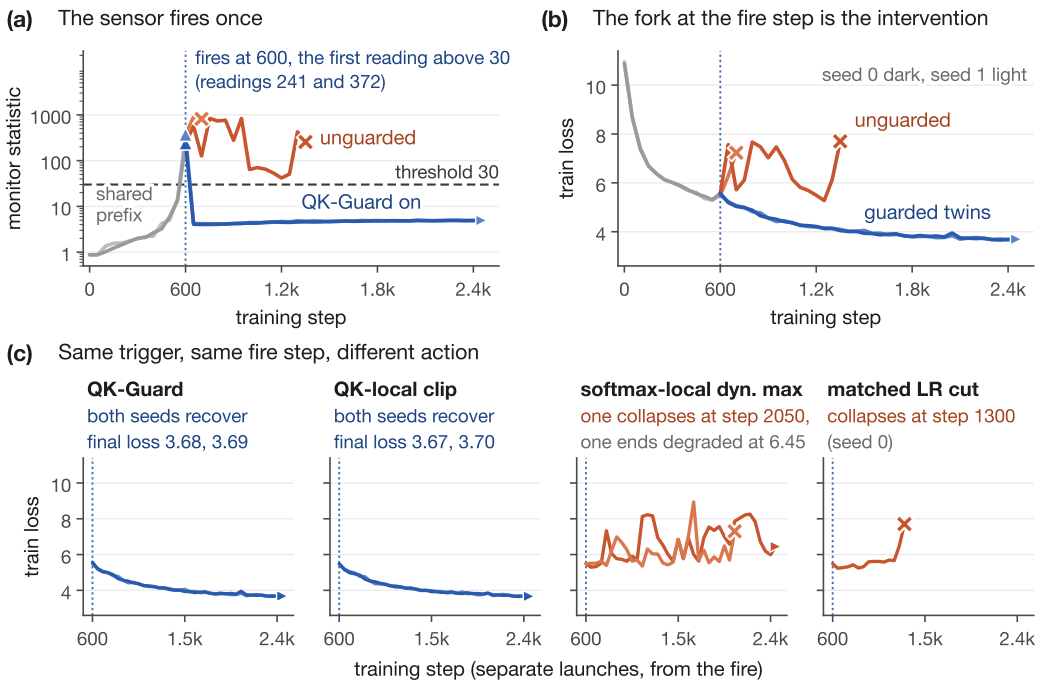

Figure 3: QK-Guard anatomy. **(a)** the monitor fires (▲) at the first periodic reading of the attention-logit statistic (Appendix A) above the fixed threshold of 30 (dashed) and switches on per-head QK normalization for good; the reading already sits above the threshold because monitoring is periodic. **(b)** unguarded runs collapse (×); their guarded twins reach the planned horizon (►) healthy; each seed-matched pair (seed 0 dark, seed 1 light) is identical through the fire step (gray), the fork at the dotted line being the intervention. **(c)** separately launched runs, shown from the fire step: at the same trigger and fire step, both QK-local actions rescue both seeds, the softmax-local dynamic maximum fails (one collapse, one degraded endpoint; ► marks censoring, never health), and the matched learning-rate cut still diverges (§5.2).

whereas all six guarded arms fire before collapse and finish healthy (Figure 3, which also shows §5.2's same-fire-step action contrast).

5.2 At a fixed trigger, the action decides rescue

A working alarm settles when to act; holding the trigger and the fire step fixed turns the choice of action into a controlled comparison. The QK-local clip, a second QK-local action distinct from the normalization that QK-Guard fires, restores healthy training on both seeds, at final train losses of 3.67 and 3.70, while the softmax-local dynamic maximum does not, one seed plateauing degraded at 6.45 and the other collapsing at 8.07; a softmax-local cap at the clip's nominal bound also rescues, but only with its derivative, which damps the gradient saturated scores send to Q and K (Appendix C). The generic controls fail in different ways: a tenfold learning-rate cut at the same fire step still diverges (6.65), rollback with reshuffling and an optimizer reset re-collapses from both post-onset restore points (8.86 and 9.40), and always-on gradient clipping was already active in every collapsed run reported here.

Online, thresholded query–key intervention already exists: Kimi K2's query–key clip measures per-head maximum logits, rescales query and key weights above a threshold, and reports an ablation of that threshold (Kimi Team et al., 2025), and a recent characterization likewise intervenes mid-run (Su et al., 2025). QK-Guard instead permanently enables normalization after the first monitored crossing. Our evidence is comparative: at the same trigger and fire step, the softmax-local dynamic maximum fails while the QK-local clip rescues. The other interventions (Appendix C) act from step zero or on magnitude: always-on query–key normalization (Henry et al., 2020; Dehghani et al.,

2023; OLMo Team, 2024), loss scaling and rollback (Micikevicius et al., 2017; Chowdhery et al., 2022), gradient clamping (Kumar et al., 2025; Huang et al., 2025; Wang et al., 2025), and spectral learning-rate control (Qi et al., 2025); detectors report without acting (Huang et al., 2026; Qiu et al., 2025), and predictors anticipate faults (Wortsman et al., 2023; Emadi, 2026).

5.3 Transfer Across Architecture, Scale, and Fault Source

The threshold was fixed on the plain GPT-2 fault; RoPE permits a causal test of architecture transfer. Switching on the monitor at a pre-collapse checkpoint on the doomed RoPE baseline's own trajectory makes the guard fire while loss is still normal, and the run continues past the step where its monitor-off twin collapses, ending at 3.45 versus the twin's 11.21. Replaying that baseline locates the crossing at step 6150, at least 1500 steps before collapse, with loss still healthy at 3.47.

Independent RoPE launches replicate beyond the replay: an independently seeded from-scratch control fires at 8700 (maximum logit 31.8) and stays healthy to 9450 at 3.45 validation loss, while the monitor-off configuration, on its own trajectory, collapses at 6800 and ends at 11.29; a later pair, bitwise identical through the fire step, reproduces the split. Six further matched 16000-step pairs (Table 6) fire in 6 of six cases between 6100 and 13400; 6 of six unguarded twins collapse after their pair's fire step, whereas 0 of six guarded arms collapse, and every guarded arm reaches 15900 logged steps.

Scale transfer is tested at 350M with the same threshold: query–key normalization ends at 4.04, and an fp32 streaming-core hot-swap at 4.03 train and 4.02 validation loss, versus the shared unguarded arm's 8.28. Six further seeds in a same-regime configuration replicate the pattern across both actions: 4 of four untreated seeds collapse against 0 of eight rescued arms at twice the original batch, 2 of two against 0 of four batch-matched (Appendix C).

A held-out fault tests source transfer: quantizing every weight gradient after reduction replaces the tuning fault entirely, yet threshold and action carry over unchanged: both seeds fire, at steps 4600 and 5600, and finish within +0.0064 and +0.0119 nats of always-on query–key normalization, while one same-configuration reference run collapses.

5.4 Endpoint gaps and deployment durability

Across the six plain rescue arms, every raw final train and validation endpoint gap to the always-on query–key normalization reference of the same configuration is below 0.03 nats. For durability, three seeds per method run a 60k-step schedule: none of the six always-on or QK-Guard arms collapses through 59900 logged steps (validation endpoints per seed in Appendix C), whereas the three controls collapse at 7200, 8600, and 7400, so survivors run 8.32 times longer than the earliest collapse. Their peak $\sigma(W^Q)$, 46.5, far exceeds values at which unguarded runs collapse, so durability is read from the loss. Under the model-wide four-bit weight-gradient fault, one seed per arm, always-on query–key normalization reaches 199900 logged steps of a 200k schedule with σ at 65.1, and QK-Guard reaches 199900 with σ at 71.9; neither collapses.

6 Conclusion

Low-precision faults with different sources (a bf16 streaming-softmax accumulator, and four-bit weight-gradient quantization applied inside or outside attention) converge on one query–key channel: correcting only that route keeps training stable while each fault stays active, and removing the channel's own dominant directions suppresses the runaway where equal-energy removal elsewhere does not. For the injected error, temporal sign-coherence decides entry, but ordinary learning is coherent too, so a monitor watches the resulting attention-logit saturation. QK-Guard acts on that signal, normalizing the query–key projections per head, so one action contains faults that share no source. Where a source-local repair exists, such as an fp32 streaming core, it remains the direct fix; the channel is where to act for sources that lack one or keep low precision by design. Fault source is not failure channel.

AI USE STATEMENT

In this work we used generative AI tools (large language models operated as coding and writing assistants) for writing, for retrieval and discovery, for research ideation and execution, to draft sections of the paper, and to draft mathematical definitions. For writing, they polished prose and drafted sections from outlines and result tables. For retrieval and discovery, they searched for and summarized related work. For research ideation and execution, they proposed and critiqued experimental designs, wrote experiment, monitoring and analysis code, and drafted interpretations of results. For mathematical claims, they drafted the paper's formal definitions, including the logit bound behind the query–key clip in Appendix A. We did not use generative AI to generate datasets. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix A describes the models, data, optimization, seeding, hardware and software; the crash rule that decides every collapse verdict; the measurement and fault-injection protocols; the interventions and the monitor that triggers them; the provenance of the threshold of 30; and the seeds, outcomes and horizons of the arm families behind the headline results. Appendix B reports the control battery around the causal test of Section 4.2, Appendix C the per-arm intervention and structure-control results, and Appendix D further analyses. Every measured value and figure in the paper is computed by script from the stored outputs of our runs and probes. The anonymous supplementary material contains the final result data behind every reported number and figure, together with the code of our probes and analyses. It does not include the training program, which derives from the public code of Qiu & Yao (2026); that code is published without a license, so we do not redistribute it or our modified version. The material therefore supports checking and reanalyzing the reported results rather than rerunning training from scratch.

REFERENCES

Ben Anson and Laurence Aitchison. Controlling changes to attention logits. *arXiv preprint arXiv:2511.21377*, 2025.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL http://arxiv.org/abs/2204.02311v5.

Musa Cim, Sarthak Arora, Poovaiah Palangappa, Miro Hodak, Ravi Dwivedula, Meena Arunachalam, and Mahmut Taylan Kandemir. Pretraining large language models with MXFP4 on native FP4 hardware, 2026. URL https://arxiv.org/abs/2605.09825.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Tri Dao. Comment on "why low-precision transformer training fails: An analysis on flash attention". GitHub Discussion in Dao-AILab/flash-attention, 2025. URL https://github.com/Dao-AILab/flash-attention/discussions/1931. Discussion #1931; comment posted 13 October 2025; accessed 28 July 2026.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL http://arxiv.org/abs/2205.14135v2.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters, 2023. URL http://arxiv.org/abs/2302.05442.

Peijie Dong, Ruibo Fan, Yuechen Tao, Di Mou, Wenhu Hu, Zhenheng Tang, Yinghao Yu, Jiamang Wang, Wenbo Su, Guodong Yang, Liping Zhang, Xiaowen Chu, Baochun Li, and Bo Li. Dissecting outlier dynamics in llm nvfp4 pretraining, 2026. URL http://arxiv.org/abs/2602.02047.

Seyed Morteza Emadi. Rank-aware spectral bounds on attention logits for stable low-precision training, 2026. URL http://arxiv.org/abs/2602.18851v1.

Aaron Gokaslan and Vanya Cohen. OpenWebText corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Alicia Golden, Samuel Hsia, Fei Sun, Bilge Acun, Basil Hosmer, Yejin Lee, Zachary DeVito, Jeff Johnson, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. Is flash attention stable?, 2024. URL http://arxiv.org/abs/2405.02803v1.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision, 2015. URL http://arxiv.org/abs/1502.02551.

Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers, 2020. URL http://arxiv.org/abs/2010.04245v1.

Nicholas J. Higham. The accuracy of floating point summation. *SIAM Journal on Scientific Computing*, 14(4):783–799, 1993. doi: 10.1137/0914050.

Ruixuan Huang, Hantao Huang, Yifan Huang, Ansheng You, Zhenxing Zhang, and Shuai Wang. Mechanism-driven monitors for preemptive detection of llm training instability, 2026. URL https://arxiv.org/abs/2606.28116.

Tianjin Huang, Ziquan Zhu, Gaojie Jin, Lu Liu, Zhangyang Wang, and Shiwei Liu. Spam: Spike-aware adam with momentum reset for stable llm training, 2025. URL http://arxiv.org/abs/2501.06842v2.

Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Chenxiao Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Qizheng Gu, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yang Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Haoyu Lu, Lijun Lu, Yashuo Luo, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Zeyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan

Su, Lin Sui, Xinjie Sun, Flood Sung, Yunpeng Tai, Heyi Tang, Jiawen Tao, Qifeng Teng, Chaoran Tian, Chensi Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Si Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Haoning Wu, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Jing Xu, Jing Xu, Junjie Yan, Yuzi Yan, Hao Yang, Xiaofei Yang, Yi Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Siyu Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Shaojie Zheng, Longguang Zhong, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi k2: Open agentic intelligence, 2025. URL http://arxiv.org/abs/2507.20534v2.

Abhay Kumar, Louis Owen, Nilabhra Roy Chowdhury, and Fabian Güra. Zclip: Adaptive spike mitigation for llm pre-training, 2025. URL http://arxiv.org/abs/2504.02507v1.

Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. Oscillation-free quantization for low-bit vision transformers, 2023. URL http://arxiv.org/abs/2302.02210.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017. URL http://arxiv.org/abs/1711.05101v3.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training, 2017. URL http://arxiv.org/abs/1710.03740v3.

Maxim Milakov and Natalia Gimelshein. Online normalizer calculation for softmax, 2018. URL http://arxiv.org/abs/1805.02867v2.

Igor Molybog, Peter Albert, Moya Chen, Zachary DeVito, David Esiobu, Naman Goyal, Punit Singh Koura, Sharan Narang, Andrew Poulton, Ruan Silva, Binh Tang, Diana Liskovich, Puxin Xu, Yuchen Zhang, Melanie Kambadur, Stephen Roller, and Susan Zhang. A theory on adam instability in large-scale machine learning, 2023. URL http://arxiv.org/abs/2304.09871v2.

OLMo Team. 2 olmo 2 furious, 2024. URL http://arxiv.org/abs/2501.00656.

Xianbiao Qi, Yelin He, Jiaquan Ye, Chun-Guang Li, Bojia Zi, Xili Dai, Qin Zou, and Rong Xiao. Taming transformer without using learning rate warmup. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=GeUK3zGreN.

Haiquan Qiu and Quanming Yao. Why low-precision transformer training fails: An analysis on flash attention. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=0jHyEKHDyx.

Haiquan Qiu, You Wu, Yingjie Tan, Yaqing Wang, and Quanming Yao. Spectral alignment as predictor of loss explosion in neural network training, 2025. URL http://arxiv.org/abs/2510.04202v1.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI technical report, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, Stosic Dusan, Venmugil Elango, Maximilian Golub, Alexander Heinecke, Phil James-Roxby, Dharmesh Jani, Gaurav Kolhe, Martin Langhammer, Ada Li, Levi Melnick, Maral Mesmakhosroshahi, Andres Rodriguez, Michael Schulte, Rasoul Shafipour, Lei Shao, Michael Siu, Pradeep Dubey, Paulius Micikevicius, Maxim Naumov, Colin Verrilli, Ralph Wittig, Doug Burger, and Eric Chung. Microscaling data formats for deep learning, 2023. URL http://arxiv.org/abs/2310.10537.

Oleg Rybakov, Mike Chrzanowski, Peter Dykas, Jinze Xue, and Ben Lanir. Methods of improving LLM training stability. *arXiv preprint arXiv:2410.16682*, 2024.

Charbel Sakr, Naigang Wang, Chia-Yu Chen, Jungwook Choi, Ankur Agrawal, Naresh Shanbhag, and Kailash Gopalakrishnan. Accumulation bit-width scaling for ultra-low precision training of deep networks, 2019. URL http://arxiv.org/abs/1901.06588v1.

Huangyuan Su, Mujin Kwun, Stephanie Gil, Sham Kakade, and Nikhil Anand. Characterization and mitigation of training instabilities in microscaling formats, 2025. URL http://arxiv.org/abs/2506.20752.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021. URL http://arxiv.org/abs/2104.09864.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL http://arxiv.org/abs/2302.13971v1.

Guoxia Wang, Shuai Li, Congliang Chen, Jinle Zeng, Jiabin Yang, Dianhai Yu, Yanjun Ma, and Li Shen. Adagc: Enhancing llm pretraining stability via adaptive gradient clipping, 2025. URL http://arxiv.org/abs/2502.11034v3.

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, 2023. URL http://arxiv.org/abs/2309.14322v2.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Josh Susskind. Stabilizing transformer training by preventing attention entropy collapse, 2023. URL http://arxiv.org/abs/2303.06296v2.

Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019. URL http://arxiv.org/abs/1910.07467.

Peiyuan Zhang, Matthew Noto, Wenxuan Tan, Chengquan Jiang, Will Lin, Wei Zhou, and Hao Zhang. Attn-qat: 4-bit attention with quantization-aware training, 2026. URL http://arxiv.org/abs/2603.00040.

A Experimental Setup, Protocols, and Measurement Semantics

A.1 Models, data, and optimization

All main-grid experiments train GPT-2-class decoder-only transformers (Radford et al., 2019) on OpenWebText (Gokaslan & Cohen, 2019) with AdamW (Loshchilov & Hutter, 2017) ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0), learning rate 10^{-3} with 2,000 warmup steps and cosine decay toward 10^{-5} on a decay horizon fixed far beyond the length of every run reported here — so these runs end while the rate is still near its peak, not at the floor; the supplementary material gives the constant — gradient clipping at 1.0, block size 1,024, and micro-batch size 20 with gradient accumulation set per arm. The base model is GPT-2 small (12 layers, 12 heads, width 768); the variants of the main text modify one component at a time (RoPE (Su et al., 2021), RMSNorm (Zhang & Sennrich, 2019), a GPT-2M and a 350M scaling, and a LLaMA-style architecture (Touvron et al., 2023) combining them). Training is distributed data-parallel; recorded process counts vary by arm family, including two for the direction-removal battery of §4.2, four for the matched-structure assay, and eight for the operand-level arms. We take `nproc` from the per-run configurations that record it. The main text states the shape wherever it bears on a result (the two- versus eight-rank contrast of Appendix D).

The remaining protocol is the nanoGPT recipe, stated here so the grid is reproducible without reading our code. The corpus is the public OpenWebText release, split into train and validation by a shuffled split holding out a fraction 0.0005 under split seed 2357 — both fixed once and shared by

every arm, so all reported validation losses are measured on one held-out set. Text is tokenized with the GPT-2 byte-pair encoder, documents separated by the end-of-text token and packed into one flat token stream from which each micro-batch draws uniformly at random; there is no document masking, so a sequence may span a document boundary. Gradient accumulation is specified globally and divided across the ranks, and the data sampler is seeded per run rather than per rank, so every rank draws the same index stream and the ranks average numerically different copies of one gradient. Two counts therefore differ and we give both. The tokens *processed* per optimizer step are the global accumulation count times the micro-batch times the block size; the *distinct* tokens sampled per step are that quantity divided by the rank count. The global accumulation count is 16 for the main grid and 8 for the batch-matched 350M pair, so the two-rank direction-removal arms sample half, and the eight-rank operand-level arms an eighth, of the processed figure. Every load-bearing comparison in this paper is between arms at a single rank count, so the arms of a contrast always share one data stream and one distinct-token batch; what the distinction changes is what a reader must hold fixed to reproduce an arm, and how the two- versus eight-rank contrast of Appendix D should be read. Weights are initialized normal with standard deviation 0.02, except the residual output projections, which use $0.02/\sqrt{2L}$ for L layers as in GPT-2; the token embedding and the output head are tied. The objective is mean token-level cross-entropy on next-token prediction. Reported train and validation losses are that same objective averaged over freshly drawn evaluation batches from the corresponding split, with the model in evaluation mode, computed at the evaluation interval that also drives the monitor (Appendix A.4); every loss in this paper is this language-modelling loss, in nats. The evaluation batch is 16 sequences, drawn independently of the training sampler and independent of the micro-batch size, and the number of such batches averaged is the per-family evaluation-iteration count; both are fixed within an arm family, so a reported loss is comparable across the arms of any contrast.

Three random-number streams are seeded separately, and a run is identified by one integer. The global PyTorch generator, which drives initialization and dropout, is seeded with that integer plus the rank, so ranks differ where they must. The training data sampler has its own generator seeded from the run integer alone, without a rank term — this is what makes the ranks share one index stream, as described above — so which indices the stream yields depends only on the run, though how many of them an optimizer step consumes depends on the rank count, as above. The quantizer's stochastic-rounding stream is reseeded every step from the step index, so the random variates an arm draws are a deterministic function of its step; the injected error is those variates applied to the current gradient.

The main experimental grid runs on NVIDIA H20 GPUs under Linux with torch 2.10 and CUDA 12.8, distributed over the per-arm rank counts given above. The cross-hardware portability check of §3.3 of the main text runs the same training program that produces the main-grid collapses on a single NVIDIA B200 (torch 2.11.0+cu128, single GPU, no DDP), with the same software-emulated quantizer: an exact-code reproduction on a different GPU architecture.

One family departs from the AdamW recipe above. To test whether an adaptive optimizer is required for the failure, we swap AdamW for SGD and hold the rest of the arm fixed — same data, same block size, same injected fault, same crash rule. The learning rates are 3×10^{-2} and 10^{-1}, each with a clean control at the same rate; the crash steps, horizons and spectral bound in Table 4 are computed from each run's recorded loss and spectral-norm history by the same extractors used everywhere else, and those records do not carry the momentum setting or rank count. The family supports the qualitative statement of §3.3 of the main text: the untreated fault collapses under SGD too, while the matched clean control at the same learning rate does not.

A.2 The sustained-divergence crash rule

Every "collapse" verdict in the paper, main text and appendix alike, is decided by one rule applied to the logged training loss. Let m_t be the running minimum of the loss up to step t. A run's crash step is the first step whose loss exceeds $m_t + 1.5$ and after which the loss never again falls below $m_t + 0.75$. A run with no such step is a survivor and is reported with the last step it logged; an excursion that recovers is not a crash. The rule is deliberately insensitive to how long a diverged run kept logging, which also makes post-collapse maxima of any quantity non-comparable across arms — the reason doomed traces in the main-text figures stop at their crash step.

A.3 Measurement conventions and injection protocols

$\sigma(W^Q)$ denotes the spectral norm of a layer's query projection; reported values take the maximum over layers, and "max over the run" additionally takes the maximum over logged steps — at the same per-family cadence as the monitor, listed in Appendix A.4.

The attention-logit statistic the monitor reads is an aggregate, not a global maximum, and the distinction matters for anyone reimplementing the trigger. On a fixed four-example validation probe batch — never a training batch, so the measurement cannot perturb the trajectory — each layer forms the full causal logit matrix $A = QK^\top/\sqrt{d_h}$ on the same Q, K the layer just used, after RoPE where RoPE is active. For that layer we take the row maximum $\max_j A_{ij}$, the largest logit each query attends to, and average it over examples, heads, and query positions; the reported value is then the maximum of that per-layer average over layers. Averaging inside the layer reduces sensitivity to isolated outliers: a crossing means some layer's mean row maximum exceeds the threshold. Values elsewhere in the paper labelled "maximum attention logit" are this statistic.

The controlled-error assay of §2 and §4 computes, at the instrumented attention backward, both the faulty and the exact quantity, forms the error, and injects a manipulated version of that error while the forward pass runs either clean (the clean-forward surrogate) or with the natural low-precision path enabled (natural context); the two are named per arm and are not interchangeable. The manipulations used in the paper, each stated by what it does to the injected tensor: *scalar-mean-only* injects the mean of the current tile; *mean-removed* injects the error minus that scalar; *per-head DC* variants inject or remove the per-head mean, at per-tile granularity in the legacy arms and reduced over the whole pass and across ranks in the full-pass arms (the main text names which is which wherever the difference matters); *sign-scramble* redraws every element's sign; *batch permutation* permutes the error along the batch axis, preserving the batch-summed tensor at every head and sequence position; *sign randomization* assigns random global signs to whole-tensor injection events, or an independent sign per head once per optimizer step; *matched Gaussian* replaces the error by noise at the same root-mean-square magnitude. Injection magnitude is scaled by λ, with $\lambda = 1$ the natural size.

The post-reduction weight-gradient quantizer of §3 and §4 takes the materialized weight gradient after backward and reduction and before clipping and the optimizer step, and rounds it onto an MXFP4-style grid (Rouhani et al., 2023) (four-bit elements in blocks of thirty-two sharing a power-of-two scale); which module's gradient receives the grid is the arm's one moving part.

A.4 The interventions, defined

Three actions in §5 fire at the same trigger and fire step, so what separates them is exactly their definitions.

Per-head query–key normalization, the action QK-Guard switches on, root-mean-square normalizes each query and key vector over its own head dimension, independently per example, head, and token position: $q \mapsto q/\sqrt{\frac{1}{d_h}\sum_c q_c^2 + \varepsilon}$ with $\varepsilon = 10^{-6}$ and the statistic accumulated in fp32, and likewise for k. It introduces no learnable gain — this is what "parameter-free" means, and it is why a guarded arm and its unguarded twin have identical parameter counts. It is applied to Q and K before the rotary embedding and before the attention kernel dispatch, so it composes with every attention implementation and with RoPE rather than replacing either.

Query–key clipping, the second query–key-local action, instead bounds the worst case attainable logit. It uses the Cauchy–Schwarz bound $\max_{ij} A_{ij} \le \max_i \|q_i\| \max_j \|k_j\|/\sqrt{d_h}$ and, when that bound exceeds the threshold, rescales both Q and K by the square root of the ratio, leaving the bound at the threshold. It is a norm clamp, not a normalization: it acts only when the bound is exceeded and it preserves the relative geometry within Q and within K.

The softmax-local dynamic maximum, the non-query–key control, changes the shift used by the streaming softmax rather than the projections. When more than one entry of the current block row lies within 10^{-3} of that block row's maximum, the shift is doubled where the maximum is positive and set to zero where it is negative, which is the published remedy for the rounding bias in this recurrence (Qiu & Yao, 2026). It therefore acts on how the attention output is accumulated, and leaves Q, K, and the logits they produce untouched.

The monitor evaluates the statistic of Appendix A.3 on its fixed probe batch every evaluation interval, and the interval is a property of the arm family rather than a single constant: 50 steps for the plain-fault rescue arms, the original 350M triplet, and the RoPE replay and from-scratch arms; 100 steps for the six matched RoPE pairs, the six further 350M seeds, the held-out weight-gradient arms, and the 60k durability arms; 200 steps on B200. A crossing is therefore detected only at an evaluation, which is why firing values can sit far above the threshold — the 350M hot-swap of §5.3 fires at step 700 on a reading of 278, the first reading to cross, having been below the threshold at the preceding evaluation. The statistic is read on the post-update weights; once fired, the action is active from the next step and stays on for the remainder of the run.

A.5 The threshold of 30: provenance and sweeps

The transfer results in §5.3 use the threshold unchanged; this subsection gives its provenance and how the outcome responds when it moves.

The value was fixed on the plain GPT-2 accumulator fault when the closed loop was first built, from the gap between the maximum-logit envelopes that healthy runs reached and the far larger values doomed runs reached before collapsing. RoPE and 350M runs had been observed by then; every guarded run reported here was launched afterwards.

The value was not re-tuned: the RoPE, 350M, held-out all-weight-gradient and B200 transfer runs all used 30. A later sweep also tested 100, the value Kimi K2 reports at production scale (Kimi Team et al., 2025).

What moving the threshold does, at one seed. One seed of the held-out all-weight-gradient MXFP4 protocol carries the trigger at three settings spanning a factor of more than thirty, with the action and everything else held fixed. Raising the value delays the fire and does not change the outcome within the observed horizons: $\tau = 30$ fires at step 4700 at a monitored logit of 32, $\tau = 100$ at step 5300 (114), and $\tau = 1{,}000$ at step 5900 (1656) — the last only once the runaway has already carried $\sigma(W^Q)$ to 23. All three fire before this seed's untreated twin collapses at 8500, and none of the three collapses inside its own horizon: the $\tau = 100$ and $\tau = 1{,}000$ arms complete 19900 logged steps at 3.19 and 3.19 validation loss, and the $\tau = 30$ arm reaches 18300 at 3.21. At $\tau = 100$ the setting repeats across three seeds, firing at steps 4900, 5400 and 5300 with none collapsing, and the query–key clip of §5.2 substitutes for the normalization at that same value in two seeds, at 4600 and 4900.

On this seed the outcome holds from $\tau = 30$ to $\tau = 1{,}000$, and the two-seed sweep that follows covers settings from 10 to 100.

Moving the threshold across two seeds. Table 3 runs the unchanged controller at settings from 10 to 100, two seeds each, on the plain GPT-2 accumulator fault and on the held-out all-weight-gradient MXFP4 fault. A fire is the first monitored evaluation strictly above the setting, read every 50 steps on the plain fault and every 100 steps on the held-out fault. On the plain fault every setting fires at the evaluation at step 600 or at the next one, 50 steps later, and no guarded arm collapses before its last logged step, while both untreated seeds collapse, at 700 and 1200. On the held-out fault every setting fires and no arm collapses through 15900 logged steps.

A.6 Per-arm seed counts and horizons

Table 4 lists, for each arm family behind the paper's headline results, the seeds run, the outcome count, and each survivor's last logged step. Counts are events over the runs launched; the one LLaMA-architecture launch of the long-horizon sweep whose trajectory matched neither the collapse signature nor a healthy run is counted in neither class.

A survivor's entry is its last logged step, and a degraded-but-not-collapsed endpoint is recorded as a survivor with its loss. Horizons, evaluation cadence, architecture, rank count and fault protocol differ across families, so each count is read within its own family.

Route correction. The route-correction grid runs in a single process. Each step backpropagates the same weights and micro-batches through the faulty re-implementation and through the fp32-

Table 3: The fixed trigger moved from 10 to 100, two seeds per setting, with the action (per-head QK normalization) held fixed. ▲ first monitored evaluation strictly above the setting, with the monitored statistic there; × sustained-rule collapse step; ▶ last logged step without collapse; ^{c} ended by a four-hour wall-clock cap before the 5,000-step budget. Val. loss is the final validation loss. These launches are separate from those drawn in Figure 3.

SETTING	SEED	▲ FIRE (STATISTIC)		OUTCOME	VAL. LOSS
Plain GPT-2 accumulator fault, cadence 50					
10	0	600	(17.9)	▶ 3850^c	3.56
10	1	600	(89.5)	▶ 3900^c	3.65
20	0	650	(306.0)	▶ 3900^c	3.60
20	1	600	(89.5)	▶ 3900^c	3.65
30	0	650	(306.0)	▶ 3900^c	3.60
30	1	600	(89.5)	▶ 3850^c	3.59
50	0	650	(306.0)	▶ 3900^c	3.60
50	1	600	(89.5)	▶ 3900^c	3.65
100	0	650	(306.0)	▶ 3900^c	3.60
100	1	650	(620.0)	▶ 3900^c	3.66
untreated	0	—		× 1200	9.25
untreated	1	—		× 700	10.82
Held-out all-weight-gradient MXFP4, cadence 100					
10	0	1600	(10.4)	▶ 15900	3.26
10	1	1400	(10.7)	▶ 15900	3.26
20	0	4800	(21.4)	▶ 15900	3.26
20	1	5200	(21.2)	▶ 15900	3.27
50	0	4700	(85.0)	▶ 15900	3.26
50	1	5300	(66.0)	▶ 15900	3.27
100	0	4800	(228.0)	▶ 15900	3.26
100	1	5500	(102.0)	▶ 15900	3.27

core reference, writes the reference gradient into the named route — for the query–key route, the query and key rows, weights and biases, of every layer's attention input projection — and only then applies global-norm clipping and the AdamW step, so clipping and the optimizer state see the spliced gradient.

Horizons of the §3 route-correction grid. Table 2(a) gives that grid's collapse counts per cell, each read at its own horizon: on plain GPT-2 the two QK-corrected arms reach step 3400 and the V/O-corrected arm was stopped at step 1200 (the table's ‡); the 350M QK-corrected arms reach step 2800, the RoPE arms step 12800 and the LLaMA arms step 24800, the horizons at which the QK-corrected finals of §3.1 are read.

Table 4: Seed counts, outcomes, and horizons for the arm families behind the paper's headline results and the generality checks around them. "Coll." counts collapses over the runs launched; × marks collapse at the listed step(s) (a range spans earliest–latest); ▶ a survivor's last logged step, with (2) marking two seeds at that horizon. Where a row turns on a further quantity — a spectral bound, a final train or validation loss — that quantity is named in place. The optimizer-generality rows use SGD at their own learning rates (Appendix A). The §3 route-correction and source-shift grids report their per-cell seed counts directly in Table 2 of the main text; the statistics-carry route grid reports its two seeds inline in §3.1.

ARM FAMILY	COLL.	OUTCOME
Matched-magnitude structure assay (§4.1, natural context, $\lambda = 1$)		
true error	5/5	× 1000–2900
sign-scramble	0/4	▶ 7900 (2), 19900 (2)
batch permutation	3/3	× 1000–1300
matched Gaussian	0/2	▶ 11900
true error at $\lambda = 0.5$	3/3	× 1600–2000
Mean-component assay (§4.1)		
scalar mean only	0/2	▶ 9900, 8600
mean-removed	2/2	× 5100, 9200
per-head DC, natural	6/6	× 2500–11800
per-head DC, backward-only near threshold	2/4	× 5300, 6000; ▶ 9800, 9900
per-head DC removed, clean forward	0/2	▶ 9900
per-head DC removed, natural	2/2	× 1100, 1900
Sign-randomization arms (§4.1)		
whole-tensor sign randomized; grouping not recorded	0/2	▶ 8900
per-head signs per step	0/2	▶ 8900
Direction-removal battery (§4.2 and Appendix B; per source × seed)		
on-target removal, 4 arms	0/4	▶ 15900
equal-energy off-target, model-wide	2/2	× 8100, 15600
query–key	0/2	▶ 15900
query–key, 32k schedule	4/4	× 13000–27000
isotropic shrink, 4 arms	4/4	× 9000–11600
random-k removal, 4 arms	4/4	× 8300–13200
350M on-target	0/2	▶ 19900
350M off-target	2/2	× 13000, 16700

Table 4: Seed counts, outcomes, and horizons (continued).

ARM FAMILY	COLL.	OUTCOME
Closed-loop transfer replication (§5.3)		
RoPE from-scratch pairs, guarded	0/6	▶ 15900
their unguarded twins	6/6	× 7400–13900
350M at 2× batch, unguarded	4/4	× 900–1800
its rescued arms	0/8	▶ 2900
350M batch-matched, unguarded	2/2	× 1400, 2000
its rescued arms	0/4	▶ 2900
Deployment durability (§5.4; 3 seeds per method)		
unprotected	3/3	× 7200, 7400, 8600
always-on QK-norm	0/3	▶ 59900
QK-Guard	0/3	▶ 59900
Optimizer generality (§3.3; SGD in place of AdamW)		
untreated fault, LR 3×10^{-2}	2/2	× 2500, 3200
its matched clean controls	0/2	▶ 3900 (2); final train 5.65
untreated fault, LR 10^{-1}	1/1	× 1700; peak $\sigma(W^Q)$ 10.9
its matched clean control	0/1	▶ 3900; final train 4.59
Source shift at a further seed (Figure 1's arms, seed 2)		
weight-gradient quantizer on QK	1/1	× 7900
per-head QK-norm twin	0/1	▶ 19900
on MLP	1/1	× 7000
per-head QK-norm twin	0/1	▶ 19900
on V	1/1	× 7400
per-head QK-norm twin	0/1	▶ 19900
Boundary arms (Appendix D)		
operand-level GEMMs, fp32 accum	0/2	▶ 11900; registered seed's $\sigma \leq 39$

B THE QK-SUBSPACE CONTROL BATTERY

The causal claim of §4.2 is carried by one contrast — remove the update's component along the current weight's three leading query–key singular directions at restored norm, versus remove equal energy from outside that subspace at matched norm and magnitude. This appendix reports the full battery around that contrast, all at the GPT-2-small scale on the post-reduction weight-gradient sources (model-wide and query–key-only), two seeds each, 16k schedules. Figure 2 of the main text shows the primary contrast itself; the arms below surround it.

The two operations. At each optimizer step, and separately for every layer's W^Q and W^K, let u_k, v_k be the leading singular pairs of the weight before the step and ΔW the update the optimizer applied. On-target removal subtracts $q = \sum_k \langle \Delta W, u_k v_k^\top \rangle_F \, u_k v_k^\top$ over the three leading pairs and rescales the remainder to $\|\Delta W\|_F$. The off-target control instead subtracts a component s built from the remainder $\Delta W - q$ and a random unit direction orthogonal to every $u_k v_k^\top$ and to that remainder, drawn with the same per-step seed on every rank, such that $\|s\|_F = \|q\|_F$ and $\|\Delta W - s\|_F = \|\Delta W - q\|_F$, and rescales the same way; s has no component along any $u_k v_k^\top$, so the target component stays in place. On-target removal fixes only the first-order change of the three current leading singular values: the rescaling returns the removed energy to the other directions, and $\sigma(W^Q)$, the largest singular value over all directions, would register a runaway carried by any of them.

Plain top-three removal without renormalization behaves like the renormalized form: no arm collapses, three running to 15900 steps and one to 15200, with $\sigma(W^Q)$ between 11.5 and 15.1, so the renormalization is not what buys survival. Projecting the update off three *random* rank-one directions instead — matched in rank, not in removed energy, since a random direction carries almost none of the update's mass — does not suppress: all four arms collapse, between 8300 and 13200 steps, with $\sigma(W^Q)$ reaching 425 to 613. Shrinking the whole update isotropically by exactly the energy the top-three removal takes — the pure dose control — also does not suppress: all four arms collapse between 9000 and 11600 steps. The equal-energy off-target arms and the 32k re-runs are reported in the main text and Table 4; together the battery isolates *direction*, not update-energy dose, as what the suppression tracks.

C FULL PER-ARM RESULTS

This appendix collects per-arm values that the main text summarizes at the outcome level. Table 6 lists what the closed loop of §5 was tested against, grouped by the question each contrast answers, and Table 7 reports the deployment-durability endpoints behind §5.4: three seeds per method over a 60k-step schedule, where the survivors' final validation losses agree to within a few hundredths of a nat while every unprotected control collapses in the first quarter of the schedule.

Table 6's final group runs the guard on training with no fault at all. Its three arms share seed, world size, schedule and source files, and differ only in what is switched on — nothing, the guard, or always-on QK-normalisation. The unguarded twin crosses the fixed threshold of 30 without collapsing, so the guard fires on its counterpart at 5000, and the armed run ends within a few hundredths of a nat of the always-on arm.

Same trigger, same nominal bound, inside the softmax. Section 5.2's same-trigger comparison sets a query–key clip against a softmax-local action that bounds nothing. Table 5 adds softmax-local actions that do bound the scores: after firing, every attention score s inside the streaming softmax is replaced by $c\tanh(s/c)$, with $c = 30$, the clip's nominal bound, or $c = 8$, per-head QK normalization's nominal bound, and with the trigger, threshold and fire step unchanged. The cap matches only that nominal bound, not the distribution of the logits the softmax sees, and its derivative also nearly removes the gradient that saturated scores send back to the query and key. Classes were fixed before the runs, against the seed-matched QK-normalized arm at each seed's common last logged step H: restored with no sustained collapse through H and a validation-loss gap within $+0.15$ nats, failed on a sustained collapse or a gap of $+0.5$ or more, partial in between; no cell was partial. The cap at 30 restores all four seeds, as the clip does, and so does the cap at 8. A straight-through variant, which keeps the cap's forward values and omits only the tanh derivative factor, collapses on all four seeds, and the dynamic maximum, rerun here to the 5,000-step budget,

Table 5: The same trigger (threshold 30) followed by different actions, four seeds each. ▲ first monitored evaluation above the threshold, with the statistic there; validation loss and its gap to the seed's QK-normalized reference at the seed's common horizon H (restored within $+0.15$ nats and without a sustained collapse); full run: $\times$ sustained-rule collapse step, ▶ last logged step without collapse. Seeds 0–1 reuse the trigger sweep's QK-normalized and untreated arms, whose wall-clock cap sets H there.

ACTION AFTER THE FIRE	▲ FIRE (STATISTIC)	VAL. AT H	GAP AT H	FULL RUN
seed 0, common horizon $H = 3900$				
QK-normalized reference	650 (306.0)	3.60	—	▶ 3900
no action	—	—	—	× 1200
QK clip, bound 30	650 (306.0)	3.60	+0.0018	▶ 4950
tanh cap, $c = 30$	650 (306.0)	3.58	–0.0177	▶ 4950
tanh cap, $c = 8$	650 (306.0)	3.59	–0.0074	▶ 4950
straight-through, $c = 30$	650 (306.0)	11.22	+7.6249	× 1250
dynamic maximum	650 (306.0)	8.54	+4.9467	× 2000
seed 1, common horizon $H = 3850$				
QK-normalized reference	600 (89.5)	3.59	—	▶ 3850
no action	—	—	—	× 700
QK clip, bound 30	600 (89.5)	3.59	+0.0013	▶ 4950
tanh cap, $c = 30$	600 (89.5)	3.58	–0.0091	▶ 4950
tanh cap, $c = 8$	600 (89.5)	3.59	–0.0038	▶ 4950
straight-through, $c = 30$	600 (89.5)	8.93	+5.3425	× 850
dynamic maximum	600 (89.5)	8.35	+4.7645	× 1400
seed 2, common horizon $H = 4950$				
QK-normalized reference	650 (440.0)	3.52	—	▶ 4950
no action	—	—	—	× 850
QK clip, bound 30	650 (440.0)	3.51	–0.0064	▶ 4950
tanh cap, $c = 30$	650 (440.0)	3.50	–0.0182	▶ 4950
tanh cap, $c = 8$	650 (440.0)	3.50	–0.0160	▶ 4950
straight-through, $c = 30$	650 (440.0)	11.32	+7.8089	× 1000
dynamic maximum	650 (440.0)	9.37	+5.8545	× 2550
seed 3, common horizon $H = 4950$				
QK-normalized reference	600 (290.0)	3.52	—	▶ 4950
no action	—	—	—	× 1500
QK clip, bound 30	600 (290.0)	3.52	–0.0000	▶ 4950
tanh cap, $c = 30$	600 (290.0)	3.51	–0.0163	▶ 4950
tanh cap, $c = 8$	600 (290.0)	3.51	–0.0144	▶ 4950
straight-through, $c = 30$	600 (290.0)	11.05	+7.5284	× 800
dynamic maximum	600 (290.0)	8.16	+4.6425	× 1950

collapses on all four as well; Table 6 keeps the endpoints of its shorter earlier runs. At this nominal bound, then, a softmax-local transformation rescues as the query–key-local actions do, and removing the tanh derivative attenuation prevented rescue on these seeds.

C.1 POSITIONING AND FURTHER DETAILS

Related interventions. Static query–key normalization starts at step zero (Henry et al., 2020; Dehghani et al., 2023; Rybakov et al., 2024; Chameleon Team, 2024; OLMo Team, 2024; Qiu & Yao, 2026), with always-on query–key-side alternatives where it does not apply (Anson & Aitchison, 2025); loss scaling and rollback alter the training procedure rather than a location (Micikevicius et al., 2017; Chowdhery et al., 2022); magnitude controllers continuously clamp gradients (Kumar et al., 2025; Huang et al., 2025; Wang et al., 2025), adjust learning rates using spectral norms (Qi et al., 2025), or reparameterize weights to control spectral norm (Zhai et al., 2023); query–key reparameterization against weight oscillation in low-bit quantization (Liu et al., 2023) is a further QK-side alternative. Detectors report without acting (Huang et al., 2026; Qiu et al., 2025), while prediction anticipates faults rather than answering them (Wortsman et al., 2023; Emadi, 2026).

Table 6: What the closed loop was tested against, grouped by the question each contrast answers. "Coll." counts runs that ended in a sustained collapse over runs launched; the result cell gives whichever quantity that row's question turns on and names it — ▲ fire step, × collapse step, ▶ last logged step without collapse, "loss" a final train loss, "gap" a final-loss gap against the always-on reference. †The non-collapsing dynamic-max seed plateaus in a degraded state; neither seed returns to healthy training. ‡Two launches, neither collapsed; the first, stopped externally before the horizon, is counted as a censored run from its run record, so the endpoint reported is the second's.

ARM	COLL.	RESULT
Does the alarm fire before the collapse?		
plain, unguarded	2/2	loss 10.61, 9.86
plain, guarded	0/6	all fire pre-collapse
RoPE resume, monitor on	0/1	loss 3.45
monitor off (same trajectory)	1/1	loss 11.21
RoPE from scratch, monitored	0/1	▲ 8700
monitor off (matched config)	1/1	× 6800
six independent pairs, guarded	0/6	▲ 6100–13400; ▶ 15900
their unguarded twins	6/6	× 7400–13900
350M, unguarded	1/1	loss 8.28
QK-norm action	0/1	loss 4.04
fp32 streaming-core action	0/1	loss 4.03
replication at 2× batch, unguarded	4/4	× 900–1800
its rescued arms, both actions	0/8	all ▲ 600; ▶ 2900
batch-matched replication, unguarded	2/2	× 1400, 2000
its rescued arms, both actions	0/4	all ▲ 700, the recorded fire step
Same trigger, same fire step: does the action matter?		
QK-local clip	0/2	loss 3.67, 3.70
softmax-local dynamic max	1/2†	loss 6.45, 8.07
Does it carry to a source the controller never saw?		
all-Wgrad MXFP4, QK-Guard	0/2	▲ 4600, 5600; gap +0.0064, +0.0119
always-on QK-norm	0/2	the reference those gaps are taken against
What does arming the guard cost when there is no fault?		
unguarded twin	0/1	▶ 9900; loss 3.58, val 3.60
QK-Guard armed	0/1	▲ 5000; loss 3.35, val 3.37
always-on QK-norm	0/2‡	loss 3.34, val 3.35

Table 7: Deployment durability at the 60k horizon (§5.4), per seed. For the unprotected method the cell gives the three crash steps (×) under the sustained rule of Appendix A; for the survivors it gives the shared horizon (▶, all three seeds) followed by the three final validation losses. The survivors' peak $\sigma(W^Q)$ over the horizon reaches 46.5; durability is read from the loss.

METHOD	COLL.	PER-SEED OUTCOME
none	3/3	× 7200, 7400, 8600
QK-norm, always on	0/3	▶ 59900 (3); 3.13, 3.09, 3.09
QK-Guard	0/3	▶ 59900 (3); 3.13, 3.10, 3.09

Native-hardware studies report the instability and leave the channel-level account open (Cim et al., 2026; Dong et al., 2026).

Older forms of the principle. The broader principle that temporally correlated gradient estimates destabilize training is older still (Molybog et al., 2023), though not in terms of attention or of arithmetic error; prior work likewise leaves aggregate deviation's link to instability open (Golden et al., 2024).

Table 8: What the injected error's structure decides. Rows name the treatment applied, not a property of the resulting trajectory. Panel (a) holds per-injection magnitude fixed, (b) varies the component, (c) compares whole-tensor sign randomization at unchanged per-injection magnitude with per-head sign randomization at exact per-step carrier magnitude. "Coll." gives collapsed runs over runs tested; × the collapse step(s), ▶ the last logged step without collapse, (2) two seeds sharing that horizon, "bwd." backward.

INJECTED-ERROR TREATMENT	COLL.	STEP
(a) Destroy one property, at matched magnitude		
nothing (the true error)	5/5	× 1000–2900
permuted across the batch	3/3	× 1000–1300
sign scrambled per element	0/4	▶ 7900 (2), 19900 (2)
replaced by Gaussian noise	0/2	▶ 11900
(b) Inject or remove one component		
injected:		
per-tile scalar mean only	0/2	▶ 9900, 8600
the error minus that scalar	2/2	× 5100, 9200
full-pass per-head DC, natural	6/6	× 2500–11800
per-tile per-head DC, bwd.	2/4	× 5300, 6000; ▶ 9800, 9900
removed:		
per-tile per-head DC, clean fwd	0/2	▶ 9900
full-pass per-head DC, natural	2/2	× 1100, 1900
(c) Randomize its sign, at matched magnitude		
global sign randomized; grouping not recorded	0/2	▶ 8900
one sign per head, per step	0/2	▶ 8900

Why the streaming recurrence went unmodelled (§2). Low-precision accumulation was known to destabilize training (Sakr et al., 2019), but classical analysis expected shorter tiled reductions to be safer (Higham, 1993), leaving this failure mode unmodelled.

D FURTHER ANALYSES

Why no per-step summary reads the gate. The per-step error summaries we registered in advance cannot tell sign structure apart. Take the realized weight-gradient error of a collapsing run and scramble its element signs: the result agrees with the original to three decimals on both, 0.734 against 0.734 in relative ℓ_2 and 0.814 against 0.814 in ℓ_∞, while its per-head DC magnitude is the larger of the two. The constructed foil shows that two errors of opposite sign structure can agree exactly on the summaries, and that the one summary that does move between them moves the wrong way. Since these are per-step readings of the error itself, and the property the outcomes turn on is a relation between steps, this is less a gap in the particular summaries than a hint about where to look. That prior work reads such aggregate proxies (Golden et al., 2024) while leaving their link to instability explicitly open is what motivated the matched design here.

The pathwise measurement. The discriminating quantity is pathwise. We fork the run live from one mid-trajectory state at which the registered cross-step summaries all sit at their null values, and drive each fork with its own variant. Write C_v for the norm of the accumulated per-head query–key contribution divided by the sum of its per-step norms, at the layer adjacent to the runaway: a fork whose contributions cancel over time sits near the value obtained by flipping the same series' signs at random, and a fork that accumulates does not. The two arms that go on to collapse reach 0.210 and 0.204 against sign-flipped nulls of 0.068 and 0.063, and reproduce the $\sigma(W^Q)$ signature at 52.0 and 62.7. The scrambled fork cancels, at 0.027 against its own null, and survives to 3500 steps; the mean-removed fork survives as well, at 0.033, near its own null.

The cross-hardware portability check in full. Figure 4 repeats the dissociation on a single Blackwell B200 running the same training program that produces the cluster collapses, with the same software-emulated quantizer. The query–key-source run collapses at 5600 steps, with matched-step spectral magnitudes agreeing with its cluster twin.

The decisive arm moves the fault off attention on that same device. With the quantizer confined to the two MLP weight gradients and the query–key gradients left unquantized, the run still develops the query–key runaway and still collapses, at 6000 steps. Over the window before the first collapse the two sources' log spectral trajectories correlate at 0.98. Triggering per-head query–key normalization mid-run then holds that same faulted run healthy to 7800 steps at 3.22 validation loss while $\sigma(W^Q)$ keeps climbing to 16. The MLP source is never repaired and $\sigma(W^Q)$ never falls: what the intervention acts on is the attention logit, and bounding that breaks the chain from spectral growth to loss at its middle stage, leaving the growth to continue far more slowly than in the untreated twin (Figure 4). An otherwise identical run under the same execution mode, without the intervention, collapses at 5200 steps – after the intervention had already fired in its twin – so the survival is attributable to the action and not to how the run was executed.

The dose–response ordering behind §2. The injected-error continuum referenced there orders as printed: crash steps 2800, 5300, 6800, and 10100 as the magnitude falls from $\lambda = 4$ to $\lambda = 0.5$, one seed per rung, the last on an extended 20k schedule; every tested rung ends in a crash.

Where fp32 accumulation is the boundary. Quantizing both operands of all three attention GEMMs to the same four-bit grid, with fp32 accumulation retained, leaves training healthy: neither of two seeds collapses through 11900 steps, with the registered seed's $\sigma(W^Q)$ no higher than 39 (a milder effective dose than the weight-gradient arms). Four-bit arithmetic is therefore not by itself the trigger within this horizon: the observed boundary lies between operand-level three-GEMM MXFP4 with fp32 accumulation and post-reduction weight-gradient quantization.

The runaway across data-parallel shapes. At the eight-rank data-parallel shape, the unstripped weight-gradient references develop the full $\sigma(W^Q)$ runaway — to 141 (model-wide) and 72 (query–key) by their 20k horizons — without loss-collapsing inside them, and on four GPUs the MLP-source fault of §3.2 likewise did not collapse within its tested horizon. The runaway, not the crash step, is what transfers across shapes, so the main text's loss-crash statements are made at the two-rank shape that exhibits them. At a fixed global accumulation count the rank count also sets the distinct-token batch, inversely (Appendix A): the eight-rank arms sample a quarter as many distinct

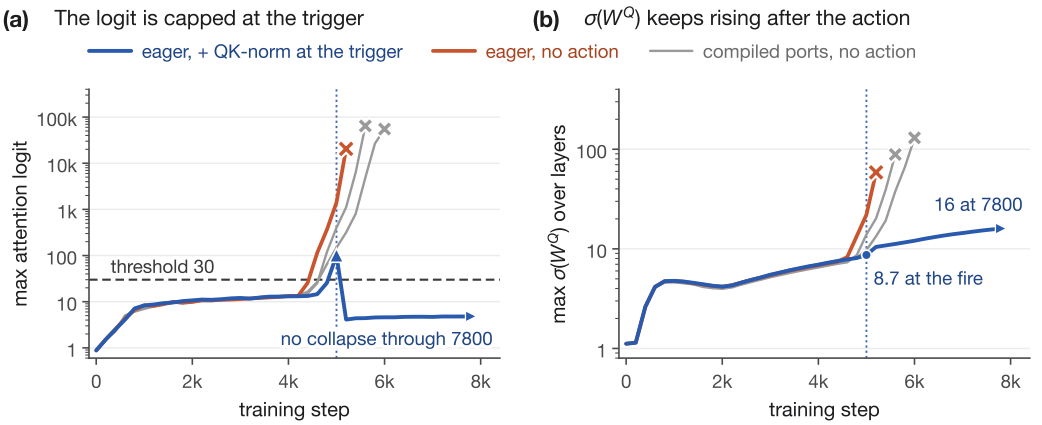

Figure 4: The B200 portability check: one B200 running our exact code, one seed per arm. **(a)** Max attention logit. The compiled QK- and MLP-source arms (gray) collapse ($\times$); in the matched eager pair the untreated twin collapses while the rescued run's logit is capped mid-run at the trigger (▲) — the threshold carried over from the H20 protocol unchanged. **(b)** Under that rescue $\sigma(W^Q)$ keeps rising: the intervention bounds the logit, not the spectral norm. Traces stop at their collapse step, so plotted height is not a severity; ► marks the scheduled horizon.

tokens per optimizer step as the two-rank arms, so step counts on the two shapes are not directly comparable.

What the two fault sources share, and what they do not. Across the two weight-gradient sources, the runaway's dominant directions overlap only partially: the squared-cosine overlap between the sources' leading runaway directions is 0.40 and 0.32 over the two seeds' established-runaway windows — far from identity, far above chance for random directions at this width. What the sources share is a state-dependent query–key-local channel, not one fixed subspace; this is the analysis behind the closing sentence of §4.2.