MECHANISM-ALIGNED OBSERVABILITY FOR TRAINING INSTABILITY OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Frontier large language model training consumes massive accelerator fleets over long wall-clock times, making stability failures extremely costly. However, global symptoms such as loss divergences often emerge only after the underlying fault has already affected the training dynamics, further increasing the sunk compute. In which internal representation can a training fault become persistently and interpretably observable before it appears in global training metrics? We propose a mechanism-aligned principle for training observability: identify the internal state into which a fault is persistently written, derive the structure induced by its mechanism, and construct monitors in representations aligned with the computation of the affected module. For FlashAttention, we show that biased backward errors accumulate into a persistent low-dimensional structure in weight updates, producing spectral concentration. For MoE routers, we characterize routing dynamics through per-token routing behavior rather than parameter similarity, which may change without altering the routing function. Experiments show that the two monitors respond selectively to changes in their corresponding modules, before degradation becomes visible in global training metrics. Our code is available at `https://anonymous.4open.science/r/mindspeed-observability-F605`.

1 INTRODUCTION

Frontier large language model (LLM) training typically occupies thousands of accelerators for weeks to months (Chowdhery et al., 2023; Smith et al., 2022). The parameter counts now reach hundreds of billions to more than a trillion (Yang et al., 2025; Kimi Team, 2025; DeepSeek-AI, 2026); pre-training corpora span tens of trillions of tokens (GLM-4.5 Team, 2025; Meituan LongCat Team, 2025; Qwen Team, 2026). DeepSeek-V3 gives an explicit cost accounting, where 14.8T pre-training tokens required 2.788M H800 GPU-hours and a reported $5.576M in direct rental cost, excluding prior research and ablation experiments (DeepSeek-AI, 2024).

At this scale, training stability has become a first-class engineering concern. Current monitors of training stability still largely rely on global symptoms such as loss collapse. GLM-130B reports loss spikes and divergence in 100B-scale training (Zeng et al., 2023); and Kimi K2 introduces MuonClip to address training instability and reports 15.5T-token pre-training with zero loss spike (Kimi Team, 2025). However, the loss collapse is not where a training fault begins, but the downstream outcome of the fault propagating until the system becomes unstable. Before loss collapse appears, a training run may already have entered an unstable state and continue silently. A useful monitoring window is therefore the stage in which a fault has begun to accumulate but has not manifested as loss collapse.

Existing training systems have begun to investigate this window, commonly monitoring gradient norms and parameter norms (Fan et al., 2024). Frontier training also uses attention entropy, maximum attention logit, and router similarity to further characterize training instability (Zhai et al., 2023; Takase et al., 2023; Golden et al., 2024; Kimi Team, 2025). However, these indicators are expensive to obtain or difficult to relate reliably to the underlying fault. Max-logit signals are difficult to expose in production FlashAttention (FA, Golden et al., 2024) because they require kernel modification or recomputation (Kimi Team, 2025). The same fault can also have different visibility across representations. Consider low-precision FA. Abnormal updates may already be written persistently into the Q/K parameters while the raw weights W remain dominated by initialization.

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.

Observing ΔW can remove this background. But FA does not use Q/K weights independently; it computes through the QK bilinear operator. Examining how the operator changes yields a representation aligned with the fault mechanism. In MoE models, cosine similarity of router weights can reveal whether parameters are becoming redundant, but healthy models can operate at different similarity levels, whereas per-token expert scores and routing distributions directly describe the function executed by routers.

This motivates a mechanism-aligned principle for monitor design in training observability. We follow the fault-propagation path to identify the internal state into which the fault first writes persistent effects; derive the structure that this accumulation should create; and observe that structure in a representation aligned with the affected module's behavior. We instantiate it in two scenarios.

- **For low-precision FA,** we propose update monitoring because low-precision backward errors first enter the model state through parameter updates. We decompose the QK operator update and monitor its singular spectrum.
- **For MoE routers,** the intended computation is discriminative, non-collapsed expert selection. We therefore monitor per-token routing entropy. Because it reads the full softmax distribution, it can capture routing concentration before downstream discrete quantities change. We further analyze how learning rate, global batch size and auxiliary loss coefficient interact through stable-winner reinforcement and accelerate routing concentration.

Our experiments show that the proposed monitors expose structural changes before global training indicators show degradation. In one FA fault-injection run, update spectra already depart from baseline $\sim$4,000 steps before the loss upturn, while loss and gradient norm are still decreasing. Router per-token entropy reveals routing concentration under changes in training hyperparameters. The different responses of the attention and router monitors provide evidence for attributing observed changes to their mechanisms, supporting early warning and diagnosis of training instability.

2 MOTIVATION

Modern training systems can already monitor many internal states, yet existing monitors can still fail to accurately reflect an ongoing fault because of two forms of mismatch: (1) fault-related changes may already be written into parameters while being obscured by the historical background; (2) a substantial change in a parameter-side metric may not correspond to a change in its actual module function. We illustrate them using low-precision FlashAttention and MoE routers to show that the observability of training instability depends on the representation being monitored and on whether that representation is aligned with the underlying fault mechanism and the module's function.

2.1 FLASHATTENTION: FAULT SIGNALS MASKED BY RAW WEIGHTS

FA fuses the softmax-scaled $QK^\top$ computation in on-chip memory and does not materialize the full $N \times N$ logit matrix. This brings dramatic memory and throughput gains, and modern LLM training and inference now rely on it as the prevalent attention implementation. However, FA is reported as a source of training instability, where low-precision arithmetic in its backward pass can deposit persistent, biased errors into weight updates (Kimi Team, 2025; Qiu & Yao, 2025). Moreover, the fused implementation blocks the most natural symptom-level monitor used at scale, namely tracking the maximum attention logit (max-logit) to catch softmax explosion (Kimi Team, 2025). Reading max-logits out of a production FA kernel requires either invasive kernel modification or a recompute pass, both unacceptable in a large training run.

Runtime training monitors are viable only for quantities that require no kernel modification or activation recomputation. For example, using weight monitors (Fang et al., 2023). However, in practice, weight-based indicators are diluted by initialization energy. The natural remaining target is the parameter update ΔW itself, which is exactly the level at which low-precision FA faults have been shown to deposit their persistent damage (Qiu & Yao, 2025).

Modern LLMs are severely over-parameterized, with intrinsic dimension far below parameter count (Jacot et al., 2018; Chizat et al., 2019; Lee et al., 2019), so a fault that perturbs along currently low-impact directions is absorbed silently. Among parameter monitors, the update ΔW is

preferable to the raw weight W on signal-to-noise grounds: singular-value statistics of W_t are diluted by initialization energy (Appendix B), whereas the increment $\Delta W_{t,\tau} = W_t - W_{t-\tau}$ removes this background and exposes the update geometry directly.

2.2 MoE Routers: Different Similarity Operating Points

Many recent LLMs above 30B parameters use MoE architectures (Meituan LongCat Team, 2025; Kimi Team, 2025; Wang et al., 2024). In a top-k MoE layer (Shazeer et al., 2017), a lightweight router selects which experts process each token. Concretely, with the router weight matrix $W_R = [w_1, \ldots, w_n]$, each token x receives expert scores $s = W_R^\top x$, a softmax produces a routing distribution, and the top-k entries dictate which of the n experts are actually used. Although W_R typically holds well below 0.1% of the parameters, the choices it makes determine which of the remaining 99.9% are used on any given token. This asymmetry between trivial parameter count and outsized influence makes the router the natural place to look for MoE instability pathologies.

The community has already used router weight similarity $\mathrm{sim}(W_R)$ for practical training monitoring (Meituan LongCat Team, 2025). However, the parameter geometry of a router does not uniquely determine its expert-selection behavior. A simple counterexample is to add the same common vector a to every expert weight, which adds $a^\top x$ to every expert score. This can increase weight similarity, while leaving both the softmax routing distribution and the top-k expert selection unchanged. Our scan of open-source MoE checkpoints further shows that even well-functioning models can operate at very different values of $\mathrm{sim}(W_R)$. Router columns in the GPT-OSS family are nearly orthogonal, whereas Qwen3.5-35B-A3B reaches ~ 0.51. Table 1 further shows that markedly different router-weight similarities can coexist with nearly identical normalized routing entropies. This motivates us to focus on the relative expert scores that directly determine expert selection.

Model	GPT-OSS 20B	GPT-OSS 120B	GLM-4.7 Flash	Ling-3.0 tiny	Qwen3 30B-A3B	Qwen3.5 35B-A3B	GLM-5.3 Flash	DS-V4-Flash preview	DS-V4-Flash 0731
$\mathrm{sim}(W_R)$	-0.021 ± 0.007	0.000 ± 0.005	0.269 ± 0.027	0.225 ± 0.044	0.492 ± 0.024	0.505 ± 0.080	0.163 ± 0.026	0.180 ± 0.072	0.179 ± 0.072
$\bar{H}/\log n$ (%)	90.0 ± 3.7	94.4 ± 1.3	79.4 ± 3.2	77.8 ± 2.3	87.9 ± 2.4	94.4 ± 1.3	85.7 ± 3.8	49.3 ± 19.4	49.8 ± 20.0

Table 1: Router-weight similarity and normalized per-token entropy across open-source MoE checkpoints, reported as mean±std across router layers. Entropy is first averaged over prefill tokens within each layer; see Figure 5 for the probe and layer distributions.

3 Mechanism-Aligned Monitor Design

We adopt a mechanism-aligned principle for training monitor design. For a specific training fault, we: (1) identify the internal state into which the fault is persistently written; (2) derive the structure that this persistent effect should induce from the underlying fault mechanism; and (3) construct a monitor in a representation aligned with the actual computation or behavior of the affected module. Through this process, each monitor is tied to an explicit fault mechanism and an interpretable structure. We instantiate this principle below for low-precision FA and MoE routers, showing how different failure mechanisms lead to different internal observables.

3.1 Attention Update Monitoring

As discussed in Section 2.1, raw weights can obscure recently written fault signals under historical parameter backgrounds. Based on the known low-precision FA failure mechanism, we therefore introduce the windowed parameter update $\Delta W_{t,\tau} = W_t - W_{t-\tau}$ as the first monitoring representation. Qiu & Yao (2025) shows that low-precision FA can induce biased scalar errors in $\mathrm{rowsum}(dO \odot O)$, and that these biased scalars multiply structurally coherent rank-one update atoms. We next ask what observable structure this mechanism should leave in attention updates over a monitoring window.

Let $X \in \mathbb{R}^{N \times d}$ be the input entering the query projection, $K = XW_k$, and $P = \mathrm{softmax}(\frac{QK^\top}{\sqrt{d_k}})$ be the attention probability matrix. For a token-step sample j, let X_j and $(PK)_j$ denote the correspond-

ing rows. Following Qiu–Yao's mechanism, the update-side source can be written as $e_j R_j$, where e_j is the biased scalar error induced through $\delta_{\mathrm{FA}} = \mathrm{rowsum}(dO \odot O)$ and $R_j = X_j^\top (PK)_j \in \mathbb{R}^{d\times d_k}$ is the associated rank-one update atom, up to the attention scale and sign. If $M = \mathbb{E}[e_j R_j]$ has rank $r \ll d_k$ and the centered fluctuations $e_j R_j - M$ are independent (or martingale-difference) with bounded second moment, then over n samples

$$\sum_{j=1}^{n} e_j R_j = nM + O_p(\sqrt{n}). \tag{1}$$

The coherent low-rank component grows linearly, while zero-mean residuals grow sublinearly. Once $n\|M\|_2$ dominates the residual, the update becomes concentrated in the low-dimensional subspace characterized by M. This turns Qiu–Yao's per-step source model into a prediction that accumulated ΔW spectra should develop a low-rank component. Its proof is in Appendix C.

Quantifying Spectral States. The structural state of ΔW can be summarized by various mathematical quantities. Given its singular values $\sigma_1 \geq \cdots \geq \sigma_r$, the *stable rank* $\mathrm{srank}(\Delta W) = \frac{\|\Delta W\|_F^2}{\|\Delta W\|_2^2}$ (Ipsen & Saibaba, 2024; Roy & Vetterli, 2007) measures the ratio of the squared Frobenius norm to the squared spectral norm, which is the inverse fraction of squared spectral energy carried by the largest singular value. However, it cannot distinguish how the remaining energy is distributed across the spectrum. On the other hand, effective rank $\mathcal{S}_\alpha(\Delta W) = \exp\big(-\sum_i p_i \log p_i\big)$ with $p_i = \frac{\sigma_i^\alpha}{\sum_j \sigma_j^\alpha}$ is another way of evaluating the state of the spectrum. Empirically, to balance sensitivity and noise, we use $\alpha = 2$, which is also known as *singular spectrum* (Alter et al., 2000).

The Bilinear Decomposition. The ΔW monitor above treats W_q and W_k as independent matrices and computes spectral metrics on ΔW_q, ΔW_k, or their concatenation. However, in FA, the attention score $QK^\top = X(W_q W_k^\top)X^\top$ depends on W_q and W_k only through the *bilinear form* $F(W_q, W_k) = W_q W_k^\top$, so monitoring the factors separately can miss correlated drifts that cancel or amplify in the product. A natural alternative is to track $\Delta F = F_t - F_{t-\tau}$ that can be decomposed into components with distinct physical and spectral signatures. For $W_{q,t}, W_{k,t}$ at two time points and $\Delta W_q = W_{q,t} - W_{q,t-\tau}$, $\Delta W_k = W_{k,t} - W_{k,t-\tau}$, with the un-subscripted factors evaluated at the base point $t - \tau$,

$$\Delta F := W_{q,t} W_{k,t}^\top - W_{q,t-\tau} W_{k,t-\tau}^\top = \Delta F_{\text{lin}} + R, \tag{2}$$

where $\Delta F_{\text{lin}} = \Delta W_q W_k^\top + W_q \Delta W_k^\top$ and $R = \Delta W_q \Delta W_k^\top$. This follows immediately from the bilinearity of F. In particular, $DF[(\Delta W_q, \Delta W_k)] = \Delta F_{\text{lin}}$ and $\frac{1}{2} D^2 F[(\Delta W_q, \Delta W_k)^{\otimes 2}] = R$, with all higher derivatives vanishing identically.

Exact Spectral Computation. Although ΔF, ΔF_{lin}, and R are formally $d \times d$ QK-product increments, their nonzero singular spectra can be computed exactly from small cores. For any $A, B \in \mathbb{R}^{d\times r}$ with thin QR decompositions $A = Q_A R_A$ and $B = Q_B R_B$,

$$AB^\top = Q_A (R_A R_B^\top) Q_B^\top, \tag{3}$$

so the nonzero singular values of $AB^\top$ are those of the $r \times r$ core $R_A R_B^\top$. Applied to a single attention head,

$$\begin{aligned} R &= \Delta W_q \Delta W_k^\top, \\ \Delta F_{\text{lin}} &= [\Delta W_q, W_q][W_k, \Delta W_k]^\top, \\ \Delta F &= [W_{q,t}, W_{q,t-\tau}][W_{k,t}, -W_{k,t-\tau}]^\top, \end{aligned} \tag{4}$$

with ranks at most d_k, $2d_k$, and $2d_k$, respectively. Thus monitoring does not require materializing a dense $d \times d$ product when $d \gg d_k$; it only requires the singular spectrum of a head-dimensional core. Appendix E reports the timing and numerical accuracy of this computation.

The functional increment ΔF is the direct monitoring quantity. Its decomposition provides a mechanism interpretation: ΔF_{lin} maps Q/K parameter updates into the QK computation to first order, while R captures their second-order interaction. This gives three testable predictions for our evaluation: (1) the fault-induced spectral concentration should be observable in the windowed updates; (2) expressing these updates through the QK computation should make the structure more directly visible than treating W_q and W_k independently; and (3) when R is small, ΔF_{lin} should closely track the spectral structure of the full QK change. We evaluate these predictions in Section 4.

3.2 MoE Router Monitoring

As discussed in Section 2.2, *routing dynamics* determine how the majority of parameters in MoE models are activated, making them a natural functional target for monitoring. Prior work shows that MoE training is particularly sensitive to optimization hyperparameters such as the learning rate (LR) and global batch size (GBS) (Zadouri et al., 2024; Tian et al., 2026; Vankadara et al., 2026). We therefore first identify the functional quantity that governs routing dynamics, and then derive how its evolution responds to perturbations in LR and GBS. Detailed derivations and their assumptions are provided in Appendix D.

In matrix form, let $C_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ and $W_{R,c} = W_R C_n = [w_1 - \bar{w}, \ldots, w_n - \bar{w}]$. Only this centered component changes the centered logits $\delta(x) = W_{R,c}^\top x$; the common-mode component adds the same scalar to every expert score and is removed by softmax. Token dispatch is therefore controlled by centered margins

$$m_{ij}(x) = (w_i - w_j)^\top x. \tag{5}$$

Unlike router weight similarity, these margins directly characterize the relative advantage of one expert over another, while removing the common offset ignored by softmax.

LR Amplifies Margin Feedback. Consider the training objective $\mathcal{L} = \mathcal{L}^{\mathrm{LM}} + \lambda\mathcal{R}$, where $\lambda \geq 0$ is the auxiliary loss coefficient (AUX) that controls load balancing. For fixed input representations and winner–competitor identities, one plain-SGD step has conditional mean

$$\mathbb{E}_t[\Delta m_{j^\star h}] = \eta\gamma_h(t) = \eta\big[\gamma_h^{\mathrm{LM}}(t) - \lambda\beta_h(t)\big], \tag{6}$$

where γ_h^{LM} is task-induced margin feedback and β_h is auxiliary feedback. Positive β_h opposes the current winner. At a given training state, LR scales the net feedback; a larger step amplifies margin expansion when $\gamma_h > 0$ and margin contraction when $\gamma_h < 0$.

Larger updates may promote routing concentration when the net feedback reinforces the current winner, while a subsequent shift toward margin contraction may permit entropy recovery. Appendix D.2 provides a qualitative illustration of these dynamics.

GBS Controls Margin Fluctuation. GBS also affects margin dynamics, but through a different mechanism from LR. To make the LR–GBS coupling explicit, for global batch size B write the empirical reinforcement coefficient as

$$\widehat{\gamma}_{h,B}(t) = \gamma_h(t) + \xi_{h,B}(t), \qquad \mathbb{E}[\xi_{h,B}(t)] = 0, \qquad \mathrm{Var}[\xi_{h,B}(t)] \approx \frac{\sigma_h^2(t)}{B}. \tag{7}$$

Then the stochastic component of the one-step margin update obeys

$$\mathrm{Var}[\Delta m_{j^\star h}(t)] \approx \eta^2\,\mathrm{Var}[\xi_{h,B}(t)] \approx \frac{\eta^2\sigma_h^2(t)}{B}. \tag{8}$$

Thus a smaller GBS increases the per-step margin variance at fixed LR. If the trajectory additionally enters the positive-$\gamma_h(t)$ stable-winner regime above, these noise-induced margin excursions may be reinforced over subsequent steps.

Quantifying Routing Concentration. The preceding analysis characterizes conditional margin drift and sampling fluctuations under the stated assumptions. These local relations do not by themselves establish monotone concentration or an ordering across training trajectories. To observe how the routing distributions evolve during training, we monitor the per-token entropy of the full softmax distribution before top-k dispatch:

$$H(p(x)) = -\sum_{i=1}^{n} p_i(x)\log p_i(x), \qquad p(x) = \mathrm{softmax}(W_{R,c}^\top x). \tag{9}$$

Entropy provides a continuous summary of probability concentration. It complements discrete assignment and aggregate load statistics, such as MaxVio (Wang et al., 2024), because the routing probabilities can become more concentrated while the selected expert set and aggregate assignment counts remain unchanged.

This observable measures concentration in the pre-dispatch routing distribution. In the top-k setting studied here, sustained entropy collapse toward zero indicates that the mixture weights become

dominated by a single expert, even though multiple experts are dispatched. Softmax saturation also attenuates the local sensitivity of routing probabilities to logit updates. Aggregate load balance does not exclude this state, because different tokens can saturate toward different experts. We therefore monitor sustained entropy collapse as a warning signal for routing saturation during training.

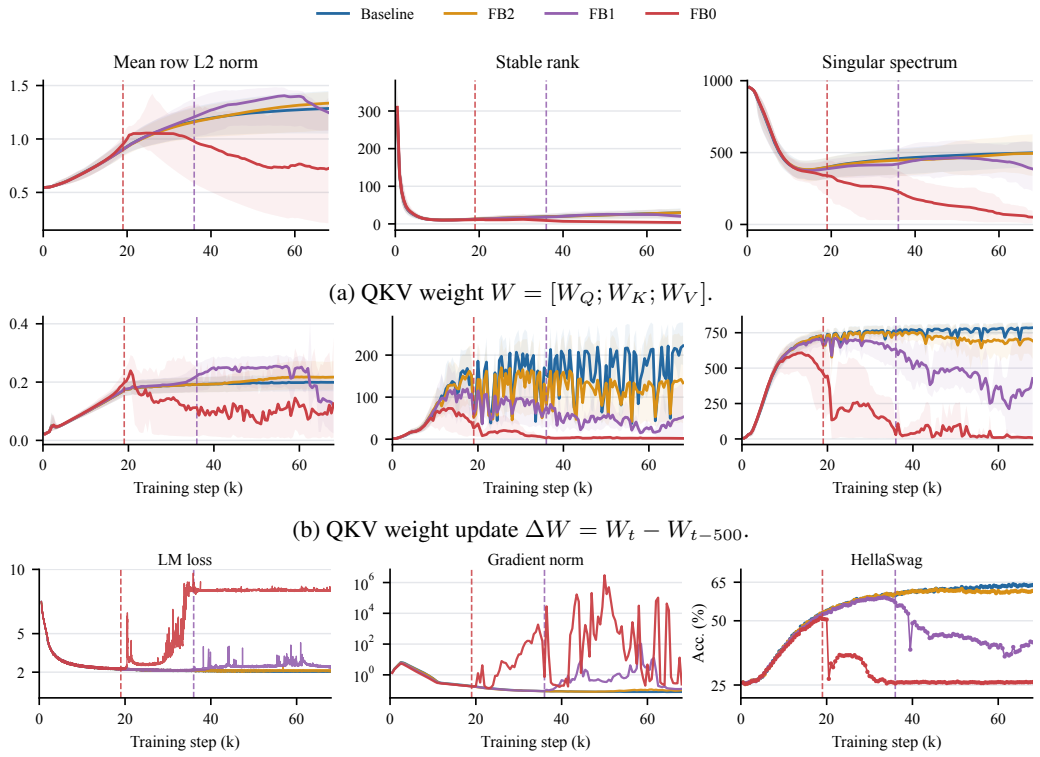
4 Experimental Evaluation

4.1 Experimental Setup

Models. We train a Qwen3-like MoE model with 8 billion total and 1 billion activated parameters per token (8B-A1B) with the proposed monitors. Its 28 layers have hidden size 1,152, and 128 routed experts with top-6 dispatch. The upper bound of the router per-token entropy is $\log 128 \approx 4.852$. This is more representative of modern LLM training than the GPT-2-small setting in prior work.

FA Fault Injection. We compare baseline training with mantissa truncation at the FA backward inputs O and dO (Li et al., 2018; Qiu & Yao, 2025). We denote each truncation setting by FBb, where b is the number of retained explicit fraction bits in both FP32 O and BF16 dO. The three settings are FB2, FB1, and FB0; Appendix C gives the implementation and error expansion.

Training Settings. For FA experiments, we use LR 5×10^{-4}, GBS 384, and auxiliary-loss coefficient (AUX) 0.001 in both baseline and fault-injected runs. For router experiments, we perturb training settings: the LR sweep uses $\{5\times10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 5\times10^{-2}\}$ with AUX $\{0, 0.001\}$ at GBS 384; the GBS sweep uses $\{24, 96, 384\}$ with LRs $\{10^{-3}, 10^{-2}\}$ at AUX 0.001.



(a) QKV weight $W = [W_Q; W_K; W_V]$.

(b) QKV weight update $\Delta W = W_t - W_{t-500}$.

(c) Training loss, gradient norm, and checkpoint evaluation.

Figure 1: (a,b) Weight statistics for baseline and FA injection settings. Curves show means across layers. Shaded bands indicate the P10–P90 range across layers. (c) Loss, smoothed gradient norm, and zero-shot HellaSwag (Zellers et al., 2019) evaluation accuracy. The vertical dashed lines mark the FB0 and FB1 loss upturns at steps ∼19,000 and 36,000.

4.2 Experiments on Low-Precision FA

The accumulation mechanism predicts that coherent arithmetic errors can concentrate the spectrum of recent attention updates before substantially changing the accumulated weights. We evaluate this in three steps: compare magnitude and spectral summaries of weights and updates, follow the updates into QK functional space, and align their trajectories with loss, gradient norm, and checkpoint evaluation. The goal is to determine whether the predicted structure is visible before overt training degradation and which representation makes that structure interpretable.

Weight vs. Its Update. Figure 1 compares baseline, FB2, FB1, and FB0 using concatenated $W = W_{QKV} = [W_Q; W_K; W_V]$. Its update is $\Delta W_t = W_t - W_{t-\tau}$, where $\tau = 500$. We compare mean row L_2 norm, stable rank, and singular spectrum. The norm measures update magnitude. Stable rank measures total spectral energy relative to the largest singular value. Singular spectrum uses the distribution across all singular values, which therefore matches the predicted redistribution of update energy more directly than a magnitude-only statistic.

In the baseline run, update spectral diversity initially grows and later levels off; a sustained reversal relative to that trajectory indicates concentration. The interpretation depends on deviation from the baseline trajectory, since healthy updates can themselves be low rank. In FB0, the update geometry departs from baseline while the raw weights remain close to it. At step 15,000, the update stable rank and singular spectrum are 51% and 87% of baseline, whereas the raw-weight values remain 96% and 97%. The update norm is 104% of baseline. Monitors on updates remove the accumulated background and expose the recent structural change.

Parameter Updates vs. QK Functional Updates. We next ask whether mapping weight updates into QK functional space makes the fault structure more apparent. To isolate the QK computation, we extract the Q/K blocks and use their concatenated update $\Delta W_{QK} = [\Delta W_Q; \Delta W_K]$ as the parameter-side reference. The first-order functional increment ΔF_{lin} expresses how these updates change the QK operator. Figure 2(a–c) shows the singular spectrum of ΔF, ΔF_{lin}, and R under baseline and FB0–FB2. Panel (d) compares the monitoring deviation of ΔF_{lin} and ΔW_{QK}.

In FB0, parameter and functional updates exhibit spectral deviations of similar magnitude, consistent with the fault structure already being visible in both representations. The benefit of ΔF_{lin} is more pronounced under the milder FB1 and FB2 settings. Figure 2(d) compares their baseline-relative spectral deficits $D = 1 - S^{\text{FB}}/S^{\text{Base}}$. Here, S denotes the singular spectrum and a larger positive deficit indicates stronger spectral concentration. At step 35,000, the deficit increases from 16.6% for ΔW_{QK} to 25.8% for ΔF_{lin} under FB1, and from 3.5% to 9.9% under FB2. These results show that QK functional updates provide more pronounced early-warning signals of training instability.

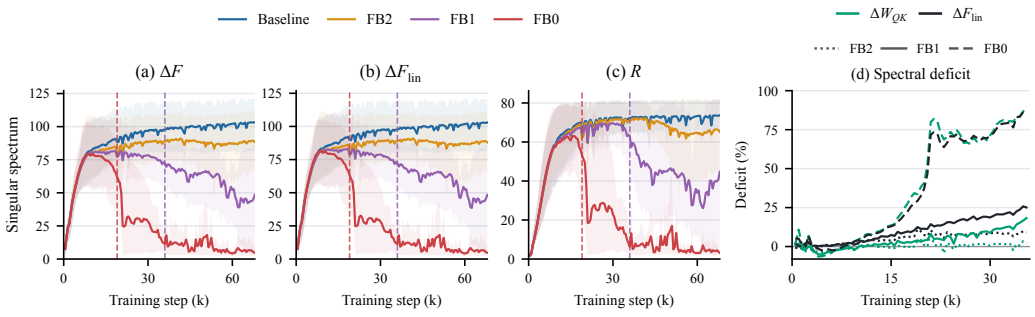


Figure 2: QK functional-update spectra and comparison with parameter updates. (a–c) Singular spectrum of ΔF, ΔF_{lin}, and R, using the same runs as Figure 1. Curves show means across layers. Shaded bands indicate the P10–P90 range across layers. (d) Baseline-relative spectral deficit D.

Preemptive Observability. Figure 1(c) places the spectral deviations in the context of training progress. Under FB0, the update spectra have already departed from baseline by step 15,000, while loss and gradient norm are still decreasing and checkpoint evaluation accuracy continues to improve. The structural warning thus precedes the loss upturn near step 19,000, despite apparently favorable training and evaluation trends. Under FB1, the functional spectral deviation is also visible before

the loss upturn at step 36,000. These observations identify a monitoring window in which structural changes are detectable before global training metrics show obvious degradation.

4.3 Experiments on MoE Routing

LR and Routing Concentration. Section 3.2 predicts that LR scales the net margin feedback, combining task-induced reinforcement with auxiliary balancing. When the net feedback favors the current winner, larger updates amplify margin separation and concentrate routing probability. Figure 3(a) shows the clearest correspondence without auxiliary balancing: across the tested LRs, higher-LR runs exhibit an earlier and deeper entropy decline and retain lower entropy throughout subsequent training. With AUX enabled, Figure 3(b) shows the LR-dependent ordering during the first 1,000 steps. Then, several runs exhibit entropy recovery, consistent with auxiliary feedback counteracting task-induced reinforcement as training evolves. However, at LR 5×10^{-2}, router per-token entropy remains low after its initial decline, showing persistent routing concentration despite auxiliary balancing. This provides early warnings of training instability and supports that auxiliary balancing can promote recovery without guaranteeing protection against excessively large updates.

GBS and Margin Fluctuations. Section 3.2 predicts that smaller batches increase the stochastic variance of margin updates at a fixed instantaneous LR. When these margin excursions reinforce the current winner, they can promote persistent routing concentration. Figure 3(c) shows that, at LR 10^{-3}, GBS 24 exhibits a more persistent entropy reduction, while GBS 96 and 384 recover to similar, higher-entropy levels. The contrast is clearer at LR 10^{-2}. As shown in Figure 3(d), GBS 384 recovers after its initial entropy decline, whereas GBS 96 and 24 develop substantially lower entropy later in training. These observations support the predicted susceptibility to routing concentration under smaller batches and larger updates, with per-token entropy exposing concentration.

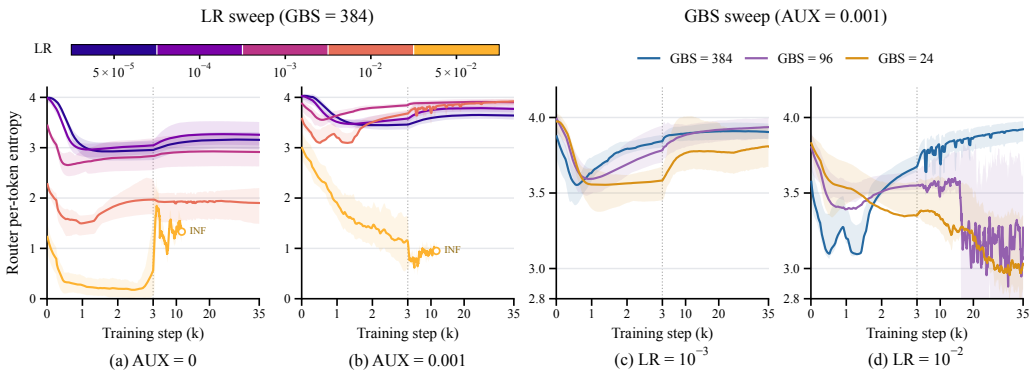


Figure 3: Router per-token entropy under LR and GBS variation. (a, b) LR sweeps. INF marks the point at which the LM loss becomes infinite and training fails. (c, d) GBS sweeps. Curves show means across layers. Shaded bands indicate the P10–P90 range across layers. Steps 0–3,000 and 3,000–35,000 occupy equal horizontal widths, each interval with a linear scale.

4.4 Selective Signatures Across Modules

A practical question for training-time monitoring is whether these responses help identify the mechanisms underlying an observed instability. This requires module-specific selectivity: each observable should preferentially reflect changes in the function it is designed to monitor. Such selectivity would strengthen the connection between the observable and its intended functional mechanism, while providing evidence for attributing an observed instability to its likely source. We therefore compare FA- and router-side responses under FA fault injection and changes in router auxiliary balancing.

The attention-update spectra separate from baseline while router entropy initially remains close to it. At step 15,000, FB0's router entropy differs from baseline by less than 0.6% (Figure 4(a)), whereas its singular spectrum ΔF_{lin} is $\sim$13% lower (Figure 2(b)). Conversely, changing AUX produces a substantially larger relative response in router entropy than in the FA monitors, see Figures 3(a,b) and 4(b). At LR 5×10^{-5} and 10^{-4}, the FA curves for AUX 0 and AUX 0.001 nearly coincide,

with NRMSEs of only 1.73% and 1.56%, respectively, whereas the corresponding router entropy NRMSEs are 15.78% and 16.13%. These results show that each monitor responds more strongly to perturbations of its target module, which helps localize the source of an observed change.

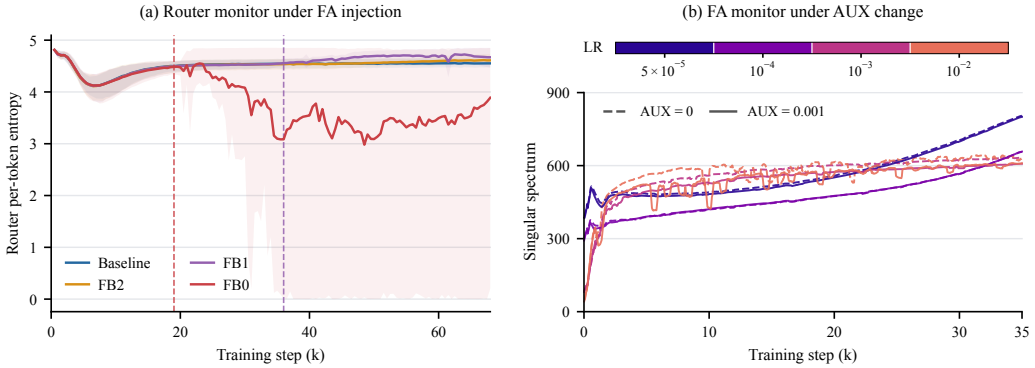


Figure 4: Cross-module responses to FA and AUX perturbations. (a) Router per-token entropy under FA fault injection; colors follow Figure 1; shaded bands indicate the P10–P90 range across layers. (b) FA singular spectrum ΔF_{lin} at GBS 384. Curves show means across layers.

5 Related Work

Attention Monitoring. Prior work diagnoses and stabilizes attention through its logits, entropy, and numerical behavior (Dehghani et al., 2023; Wortsman et al., 2024). Kimi K2 uses QK-clipping through MuonClip to prevent excessive attention logits (Kimi Team, 2025). Zhai et al. (2023) connect attention-entropy collapse to training instability and propose σReparam based on spectral normalization. Takase et al. (2023) analyze loss spikes through the spectral norms of sub-layer Jacobians. For FlashAttention, Golden et al. (2024) quantify numerical deviations and investigate their downstream effects, while Qiu & Yao (2025) identify a failure mechanism involving biased rounding errors and similar low-rank representations that corrupt weight updates. Bao et al. (2024); Pan et al. (2024) relate QK geometry to attention localization and singular-vector structure.

Router Monitoring. MoE training commonly uses auxiliary objectives to balance expert loads (Shazeer et al., 2017; Fedus et al., 2022). Wang et al. (2024) analyze interference from balancing gradients and propose loss-free balancing through dynamically adjusted expert biases. For router diagnosis, LongCat-Flash monitors pairwise cosine similarity of expert router weights and the relative gradient norms of balancing and language-modeling objectives on batch-averaged expert probabilities (Meituan LongCat Team, 2025). Behavior-side analyses also identify routing uncertainty. GW-MoE studies tokens with nearly equal expert scores and mitigates their sensitivity to expert selection (Wu et al., 2024).

6 Conclusion

We derived internal training monitors for two stability-critical modules of modern LLMs by asking what each module is supposed to compute and where damage from its known failure mechanism would first appear. For FA, biased low-precision backward errors produce coherent low-rank drift in accumulated weight updates. Monitoring the spectrum of the exact QK functional increment ΔF makes this structure more directly visible. For MoE routers, the answer is per-token entropy. Modern LLM architectures continue to introduce modules with their own internal dynamics. As these modules enter production training, each will need monitors derived from its own mechanism. The methodology demonstrated here provides a template for that derivation.

AI USE STATEMENT

In this work, we used generative AI tools only for language polishing and formatting assistance, including improving grammar, clarity, and presentation of the manuscript. We did not use generative AI tools to generate research ideas, design experiments, conduct data analysis, produce experimental results, or formulate scientific claims. All AI-assisted edits were manually reviewed by the authors to ensure that they accurately reflected our intended meaning and did not alter the technical content or conclusions of the work. We take responsibility for the final content of this work, including text or artifacts produced with the aid of generative AI.

REFERENCES

Orly Alter, Patrick O. Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97 (18):10101–10106, 2000.

Han Bao, Ryuichiro Hataya, and Ryo Karakida. Self-attention networks localize when qk-eigenspectrum concentrates. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=aRZjRj41WQ.

Lénaïc Chizat, Edouard Oyallon, and Francis R. Bach. On lazy training in differentiable programming. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 2933–2943, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/ae614c557843b1df326cb29c57225459-Abstract.html.

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. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023. URL https://jmlr.org/papers/v24/22-1144.html.

DeepSeek-AI. DeepSeek-V3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International*

Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of *Proceedings of Machine Learning Research*, pp. 7480–7512. PMLR, 2023. URL https://proceedings.mlr.press/v202/dehghani23a.html.

Haozheng Fan, Hao Zhou, Guangtai Huang, Parameswaran Raman, Xinwei Fu, Gaurav Gupta, Dhananjay Ram, Yida Wang, and Jun Huan. Hlat: High-quality large language model pre-trained on aws trainium. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 2100–2109. IEEE, 2024.

Wenyi Fang, Hao Zhang, Ziyu Gong, Longbin Zeng, Xuhui Lu, Biao Liu, Xiaoyu Wu, Yang Zheng, Zheng Hu, and Xun Zhang. A survey of metrics to enhance training dependability in large language models. In *2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 180–185. IEEE, 2023.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022. URL https://jmlr.org/papers/v23/21-0998.html.

GLM-4.5 Team. GLM-4.5: Agentic, reasoning, and coding (ARC) foundation models, 2025. URL https://arxiv.org/abs/2508.06471.

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 https://arxiv.org/abs/2405.02803.

Ilse C.F. Ipsen and Arvind K. Saibaba. Stable rank and intrinsic dimension of real and complex matrices, 2024. URL https://arxiv.org/abs/2407.21594.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8580–8589, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Kimi Team. Kimi k2: Open agentic intelligence, 2025. URL https://arxiv.org/abs/2507.20534.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8570–8581, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0d1a9651497a38d8b1c3871c84528bd4-Abstract.html.

Zhaoqi Li, Yu Ma, Catalina Vajiac, and Yunkai Zhang. Exploration of numerical precision in deep neural networks, 2018. URL https://arxiv.org/abs/1805.01078.

Bill Yuchen Lin, Yuntian Deng, Khyathi Raghavi Chandu, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=MKEHCx25xp.

Meituan LongCat Team. LongCat-Flash technical report, 2025. URL https://arxiv.org/abs/2509.01322.

Xu Pan, Aaron Philip, Ziqian Xie, and Odelia Schwartz. Dissecting query-key interaction in vision transformers. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/6216515a5e0b3257c49dcb1647e497d1-Abstract-Conference.html.

Haiquan Qiu and Quanming Yao. Why low-precision transformer training fails: An analysis on flash attention, 2025. URL https://arxiv.org/abs/2510.04212.

Qwen Team. Qwen3.5-Omni technical report, 2026. URL https://arxiv.org/abs/2604.15804.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *European Signal Processing Conference (EUSIPCO)*, pp. 606–610, 2007.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530b, a large-scale generative language model, 2022. URL https://arxiv.org/abs/2201.11990.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. Spike no more: Stabilizing the pre-training of large language models, 2023. URL https://arxiv.org/abs/2312.16903.

Changxin Tian, Kunlong Chen, Jia Liu, Ziqi Liu, Zhiqiang Zhang, and Jun Zhou. Hyperparameter scaling laws across moe sparsity, 2026. URL https://arxiv.org/abs/2609.08690.

Leena Chennuru Vankadara, Moritz Haas, Luke Hayward, Sebastian Bordt, and Alessandro Breccia. How to scale mixture-of-experts: From mup to the maximally scale-stable parameterization, 2026. URL https://arxiv.org/abs/2605.14200.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts, 2024. URL https://arxiv.org/abs/2408.15664.

Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie E. Everett, Alexander A. 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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=d8w0pmvXbZ.

Haoze Wu, Zihan Qiu, Zili Wang, Hang Zhao, and Jie Fu. GW-MoE: Resolving uncertainty in MoE router with global workspace theory, 2024. URL https://arxiv.org/abs/2406.12375.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=EvDeiLv7qc.

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

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. GLM-130B: an open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=-Aw0rrrPUF.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M. Susskind. Stabilizing transformer training by preventing attention entropy collapse. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40770–40803. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhai23a.html.

APPENDIX

A FORMULAS

Table 2 summarizes computational formulas not explicitly provided in the main text.

Table 2: Computation of the main observables and supporting quantities.

Quantity	Definition or computation
FA backward quantities	$dO = \partial\mathcal{L}/\partial O, \quad dP = \partial\mathcal{L}/\partial P = dOV^\top$; $\delta_{\mathrm{FA}} = \mathrm{rowsum}(dO \odot O)$, one scalar per token.
Mean row L_2 norm	$N(A) = \dfrac{1}{m}\sum_{j=1}^{m} \|A_{j,:}\|_2$.
Router weight similarity	$\mathrm{sim}(W_R) = \dfrac{1}{n(n-1)}\sum_{i\neq j} \dfrac{w_i^\top w_j}{\|w_i\|_2\|w_j\|_2}$, for nonzero w_i.
NRMSE	$\mathrm{NRMSE}(a,b) = \sqrt{\dfrac{\sum_t (b_t - a_t)^2}{\sum_t a_t^2}}$.

B INITIALIZATION DOMINANCE AND RAW-WEIGHT MONITORING

We justify the initialization-dominance claim used in Section 3.1. Let $W_t = W_0 + E_t$, where E_t is the total displacement from initialization. Weyl's singular-value perturbation inequality gives, for every singular index i,

$$|\sigma_i(W_t) - \sigma_i(W_0)| \le \|E_t\|_2 \le \|E_t\|_F. \tag{10}$$

Thus, when $\|E_t\|_F \le \varepsilon\|W_0\|_F$ with $\varepsilon < 1$, every singular-value statistic of the raw weight is observed through an initialization-dominated background. This does not say that the network is not learning: in lazy or NTK-like regimes, function values can change while parameter displacement remains small. It only says that a raw-weight monitor has poor signal-to-noise for faults that first alter the update geometry. The update increment $\Delta W_{t,\tau} = W_t - W_{t-\tau}$ removes W_0 exactly, so spectral concentration in the update is not diluted by initialization energy.

C ACCUMULATION CONSEQUENCE FOR FA

This section proves the concentration form of the accumulation consequence (Equation 1) used for the monitoring window. The source model follows the per-step mechanism of Qiu & Yao (2025): low-precision FA supplies biased scalar coefficients e_j multiplying coherent rank-one update atoms $R_j = X_j^\top (PK)_j$.

Index token-step samples by j and set $R_j = X_j^\top (PK)_j \in \mathbb{R}^{d\times d_k}$. Assume $Y_j = e_j R_j - M$ are independent or martingale-difference fluctuations with $M = \mathbb{E}[e_j R_j]$ and $\mathbb{E}\|Y_j\|_F^2 \le \nu^2$. Then

$$A_n := \sum_{j=1}^{n} e_j R_j = nM + Z_n, \qquad Z_n := \sum_{j=1}^{n} Y_j. \tag{11}$$

By orthogonality of the centered increments,

$$\mathbb{E}\|Z_n\|_F^2 \le n\nu^2. \tag{12}$$

Markov's inequality implies $\|Z_n\|_F = O_p(\sqrt{n})$, hence $\|Z_n\|_2 = O_p(\sqrt{n})$. This proves Equation 1. The biased mean nM grows linearly, while the zero-mean residual grows sublinearly.

Finally, suppose M has rank r and singular gap $\sigma_r(M) > 0$. Weyl's inequality gives

$$|\sigma_i(A_n) - n\sigma_i(M)| \le \|Z_n\|_2. \tag{13}$$

When $n\sigma_r(M) \gg \|Z_n\|_2$, the top r singular values of A_n are controlled by M and the remaining singular values are residual-scale. Consequently, stable rank and singular spectrum of the accumulated update converge toward those of the low-rank mean M. This is the formal sense in which biased low-precision arithmetic becomes a low-rank ΔW fault under coherence of rank-one atoms.

FA fault injection. The injection clears low-order fraction bits by reinterpreting each selected floating-point value as an integer storage word u of the same width, applying a bit mask, and reinterpreting the result in the original dtype. For a mask width n, the stored-bit operation is

$$\tilde{u} = u \,\&\, \mathrm{mask}(n), \qquad \mathrm{mask}(n) = \underbrace{11\cdots11}_{\text{retained bits}}\underbrace{00\cdots00}_{n\text{ bits}}. \tag{14}$$

In our training runs, O is retained in FP32 and dO in BF16 at the injection sites. FBb, for $b \in \{2, 1, 0\}$, uses $n_O = 23 - b$ and $n_{dO} = 7 - b$, leaving b explicit fraction bits in both tensors while preserving sign and exponent. FB0 removes all explicit fraction bits; it does not remove the implicit leading bit of normalized values. FB2, FB1, and FB0 correspond to the original log names O21/dO5, O22/dO6, and O23/dO7, respectively. Clearing fraction bits truncates finite values toward zero and creates deterministic, input-dependent error; it does not implement round-to-nearest. Reduced-precision truncation is a numerical emulation method used by Li et al. (2018). The selection of backward FA tensors is motivated by Qiu & Yao (2025), but the exact masks here are our controlled stress test, not a protocol established by that work. For a BF16 example with $n = 3$,

$$\begin{aligned} \text{original value } 1.1015625 \quad &: \quad \texttt{0|01111111|0001101}_{\text{BF16}} = \texttt{0x3f8d} \\ &\quad\;\; \downarrow \texttt{(uint16 >> 3) << 3} \\ \text{truncated value } 1.0625 \quad &: \quad \texttt{0|01111111|0001000}_{\text{BF16}} = \texttt{0x3f88}. \end{aligned} \tag{15}$$

The exact tensor and mask width are experimental knobs, but the resulting tensor-level error has the same algebraic role. If the backward path uses truncated tensors $\widehat{O} = O + E_O$ and $\widehat{dO} = dO + E_{dO}$, the scalar source can include perturbations to both O and dO:

$$\widehat{\delta}_{\text{FA}} - \delta_{\text{FA}} = \mathrm{rowsum}(dO \odot E_O + E_{dO} \odot O + E_{dO} \odot E_O). \tag{16}$$

This implementation-specific expansion only changes what contributes to the scalar e_j. Perturbing dO also induces a direct perturbation of $dP = dOV^\top$, so the main text uses only the abstract source form $e_j R_j$ rather than treating the O, dO expansion as a separate mechanism.

C.1 Magnitude Regime of QK Functional Increments

With the un-subscripted factors evaluated at $t - \tau$, bilinearity gives

$$\begin{aligned} \Delta F &= (W_q + \Delta W_q)(W_k + \Delta W_k)^\top - W_q W_k^\top \\ &= \underbrace{\Delta W_q W_k^\top + W_q \Delta W_k^\top}_{\Delta F_{\text{lin}}} + \underbrace{\Delta W_q \Delta W_k^\top}_{R}. \end{aligned} \tag{17}$$

Let $w = \max(\|W_q\|_F, \|W_k\|_F)$ and $u = \max(\|\Delta W_q\|_F, \|\Delta W_k\|_F)$. Submultiplicativity and the triangle inequality imply

$$\begin{aligned} \|\Delta F_{\text{lin}}\|_F &\le \|\Delta W_q\|_F \|W_k\|_2 + \|W_q\|_2 \|\Delta W_k\|_F \le 2wu, \\ \|R\|_F &\le \|\Delta W_q\|_F \|\Delta W_k\|_2 \le u^2. \end{aligned} \tag{18}$$

Thus the first-order and second-order terms have upper bounds of order wu and u^2, respectively. Upper bounds alone do not establish dominance: the two first-order contributions can cancel, or their products can be small. To make the magnitude regime explicit, assume $\|\Delta F_{\text{lin}}\|_F \ge cwu$ for a constant $c > 0$ independent of u/w. This condition rules out near cancellation or degenerate first-order action. Then

$$\frac{\|\Delta F - \Delta F_{\text{lin}}\|_F}{\|\Delta F_{\text{lin}}\|_F} = \frac{\|R\|_F}{\|\Delta F_{\text{lin}}\|_F} \le \frac{u}{cw}. \tag{19}$$

In the early-to-mid training regime $u \ll w$, the exact increment is therefore well approximated by its first-order term under this condition. Moreover, singular-value perturbation gives

$$|\sigma_i(\Delta F) - \sigma_i(\Delta F_{\text{lin}})| \le \|R\|_2 \le u^2. \tag{20}$$

This explains why the exact increment and its linear term can show similar spectral behavior, without guaranteeing identical values of a normalized spectral statistic or identical detection times. The second-order interaction can still have a structured spectrum despite its smaller magnitude; its role is to characterize Q/K update coupling within the decomposition. The empirical comparison in Section 4.2 uses ΔF as the direct monitor and ΔF_{lin} and R to interpret that monitor.

D THE ROUTER WEIGHT–ENTROPY LINK

This appendix connects router updates to margins and per-token entropy, including auxiliary balancing and batch noise. We analyze plain SGD with LR $\eta > 0$ on a fixed probe distribution. For $n \geq 2$ experts, let $W_R = [w_1, \ldots, w_n] \in \mathbb{R}^{d\times n}$, $C_n = I_n - \mathbf{1}\mathbf{1}^\top/n$, and $W_{R,c} = W_R C_n$. The logits are $s_i(x) = w_i^\top x$, their centered vector is $\delta(x) = W_{R,c}^\top x$, and $p(x) = \mathrm{softmax}(\delta(x))$. Centering leaves softmax unchanged. We use $H(p) = -\sum_i p_i \log p_i$ with natural logarithms, and write $\mathbb{E}_t$ for expectation conditional on the state before sampling a minibatch.

Local weight–entropy relation. For centered logits near zero,

$$H(\mathrm{softmax}(\delta)) = \log n - \frac{\|\delta\|_2^2}{2n} + R_H(\delta), \qquad R_H(\delta) = O(\|\delta\|_2^3). \tag{21}$$

Writing $H = \log Z - p^\top\delta$ with $Z = \sum_i e^{\delta_i}$, Taylor expansion at centered $\delta = 0$ gives zero gradient and Hessian $-C_n/n$, yielding Equation 21. For fixed probe second moment $M_x = \mathbb{E}_x[xx^\top]$, taking expectations gives

$$\log n - \mathbb{E}_x H(p(x)) = \frac{1}{2n}\,\mathrm{tr}(W_{R,c}^\top M_x W_{R,c}) + O(\mathbb{E}_x\|\delta(x)\|_2^3). \tag{22}$$

This assumes probe logits in the local neighborhood and finite displayed moments. For $m_{ij} = s_i - s_j$, centering gives $\sum_{i<j} m_{ij}^2 = n\|\delta\|_2^2$; hence the quadratic term also equals $\sum_{i<j}\mathbb{E}_x m_{ij}^2/(2n^2)$. This connects entropy to the energy of all margins without assuming isotropy. The expansion cannot be extrapolated to saturation: the true entropy deficit is bounded by $\log n$, whereas the quadratic term can grow without bound.

D.1 HOW A LARGER LR CAN REINFORCE THE WINNER

This section derives the mechanism behind *LR Amplifies Margin Feedback* (Section 3.2). By *winner reinforcement*, we mean an increase in the logit gap between the currently highest-scoring expert and a competitor. A larger LR amplifies this increase when the combined task and auxiliary feedback favors the winner at a fixed training state. We first derive these two feedback contributions, then connect their net effect to the winner's routing probability.

Task and auxiliary contributions to winner reinforcement. Let $\mathcal{L}_{\mathcal{B}}^{\mathrm{LM}} = M^{-1}\sum_{u\in\mathcal{B}}\ell_u$ be the LM loss over M tokens and $\mathcal{L}_{\mathcal{B}} = \mathcal{L}_{\mathcal{B}}^{\mathrm{LM}} + \lambda\mathcal{R}_{\mathcal{B}}$, with $\lambda \geq 0$. For the dense-softmax relaxation $y(x) = \sum_i p_i(x)E_i(x)$ (Jacobs et al., 1991; Shazeer et al., 2017), with expert outputs $E_i(x)$, define $q_i = M\langle\partial\mathcal{L}_{\mathcal{B}}^{\mathrm{LM}}/\partial y, E_i(x)\rangle$ and $\bar{q} = \sum_i p_i q_i$. Holding expert outputs fixed, the chain rule gives $r_i^{\mathrm{LM}} := M\,\partial\mathcal{L}_{\mathcal{B}}^{\mathrm{LM}}/\partial s_i = p_i(q_i - \bar{q})$. Thus q_i is a local loss sensitivity, and a current winner need not receive reinforcing task feedback.

For the auxiliary-specific results, assume the stop-gradient load-balancing objective

$$\mathcal{R}_{\mathcal{B}} = n\sum_i \mathrm{sg}(f_i)P_i, \qquad P_i = \frac{1}{M}\sum_{x\in\mathcal{B}} p_i(x), \qquad f_i = \frac{1}{Mk}\sum_{x\in\mathcal{B}}\mathbf{1}\{i\in\mathrm{TopK}(x)\}. \tag{23}$$

Exactly k assignments per token imply $\sum_i f_i = 1$ and $f_i \leq 1/k$. Loads are recomputed each forward pass and held fixed during differentiation. Writing $\bar{f}_p = \sum_i p_i f_i$, differentiation gives

$$a_i := M\frac{\partial\mathcal{R}_{\mathcal{B}}}{\partial s_i} = np_i(f_i - \bar{f}_p), \qquad r_i = r_i^{\mathrm{LM}} + \lambda a_i, \qquad \widehat{G} = \nabla_{W_R}\mathcal{L}_{\mathcal{B}} = \frac{1}{M}\sum_{x\in\mathcal{B}} x\,r(x)^\top. \tag{24}$$

Both logit-gradient components sum to zero, so $\widehat{G}C_n = \widehat{G}$. For balancing groups $\mathcal{B}_g$ of size M_g, use $\mathcal{R} = \sum_g (M_g/M)\mathcal{R}_g$ and each token's group loads in a_i; the same $1/M$ normalization then holds. The balancing group need not equal the global batch.

How LR amplifies the winner's advantage. Fix a probe x' and winner–competitor identities $j^\star, h$, with margin $m_{j^\star h} = (w_{j^\star} - w_h)^\top x'$. One SGD step gives

$$\Delta m_{j^\star h} = -\frac{\eta}{M}\sum_{x\in\mathcal{B}}(x'^\top x)(r_{j^\star} - r_h) = \eta\widehat{\gamma}_h, \qquad \mathbb{E}_t\Delta m_{j^\star h} = \eta(\gamma_h^{\mathrm{LM}} - \lambda\beta_h) =: \eta\gamma_h. \tag{25}$$

Here γ_h^{LM} and β_h are the conditional means of the corresponding negative task-gradient and positive auxiliary-gradient batch averages, respectively, including $1/M$ but excluding η and λ. Positive β_h opposes the winner, but an underloaded winner can instead receive auxiliary reinforcement. Thus, when $\gamma_h^{\mathrm{LM}} > \lambda\beta_h$, increasing LR at the same state produces a larger conditional mean increase in the winner's margin. Across LR runs, the net feedback can change sign and evolve differently, so this one-step relation does not order entire trajectories. For fixed ordering, strongest competitor $h^\star$, and constant LR, write $m_t = m_{j^\star h^\star}(x', t)$; then

$$\begin{aligned} m_T &= m_0 + \eta \sum_{t<T} \gamma_{h^\star}(t) + \eta \sum_{t<T} \xi_{h^\star}(t), \\ \xi_h &:= \widehat{\gamma}_h - \gamma_h, \quad \mathbb{E}_t \xi_h = 0, \quad 1 - p_{j^\star}(T) \le (n-1) e^{-m_T}. \end{aligned} \tag{26}$$

The bound follows by factoring the winner's exponential out of the softmax denominator. Positive conditional mean feedback does not by itself imply positive realized cumulative margin growth, because the accumulated sampling fluctuations can be negative. The recurrence therefore describes realized margin changes without guaranteeing monotone concentration along stochastic training trajectories.

Cyclic permutations of a probability vector with distinct entries and maximum approaching one give $f_i = P_i = 1/n$ and $a = 0$, while every token's entropy approaches zero; load balance therefore supplies no per-token entropy floor.

D.2 Transient Recovery and a Finite-Step Stability Boundary

When loads are balanced, the auxiliary derivative vanishes; subsequent load and task-gradient changes can reverse the net feedback. To illustrate recovery, assume a probe follows a fixed centered, nonconstant direction $\delta_t = z_t v$, $z_t \ge 0$. Along this direction, $dH/dz = -z \operatorname{Var}_{i\sim p}(v_i) \le 0$. Suppose task reinforcement is $A\rho^t + a_1 z_t$ and auxiliary restoration is $\lambda b z_t$, with $A, b > 0$, $a_1 \ge 0$, $0 < \rho < 1$, and $\kappa := \lambda b - a_1 > 0$. Then

$$z_{t+1} = (1 - \eta\kappa) z_t + \eta A \rho^t, \qquad z_0 = 0, \qquad z_t = \eta A \sum_{j=0}^{t-1} (1 - \eta\kappa)^j \rho^{t-1-j}. \tag{27}$$

These are assumptions on the update field with loads recomputed, not derivatives through stop-gradient or consequences of the auxiliary objective alone. For $0 < \eta\kappa < 1$, z_t initially increases and eventually returns to zero, permitting an entropy valley and recovery. With $\lambda = a_1 = 0$, instead $z_t = \eta A(1 - \rho^t)/(1 - \rho)$; adding persistent forcing $a_\infty \ge 0$ to the stable model gives $z_\infty = a_\infty/\kappa$, allowing recovery short of uniformity. For common A, κ, ρ, concentration at step $t \ge 1$ increases with LR if $t\eta\kappa \le 1$: each summand's factor $\eta(1 - \eta\kappa)^j$ has nonnegative derivative, with the $j = 0$ derivative positive. This sufficient condition does not predict the observed transition time.

Perturbations satisfy $e_{t+1} = (1 - \eta\kappa) e_t$, hence contract exactly when $0 < \eta\kappa < 2$. The valley above uses $\eta\kappa < 1$; the broader contraction condition illustrates a step-size limit, without identifying the empirical critical LR or proving full-model divergence.

D.3 GBS, Margin Fluctuations, and Entropy

We derive the conditional variance law used in *GBS Controls Margin Fluctuation* (Section 3.2) and its local entropy consequence.

Conditional margin variance. For a fixed probe and expert pair, let GBS B count sequences of fixed length L, so $M = BL$, and write $\widehat{\gamma}_{h,B} = B^{-1} \sum_b Y_{h,b,B}$, where $Y_{h,b,B}$ is the negative margin-gradient sum within sequence b, divided by L. For this total task–auxiliary feedback estimate (Equation 25), define at a common fixed state $\gamma_{h,B} = \mathbb{E}_t \widehat{\gamma}_{h,B}$ and $\xi_{h,B} = \widehat{\gamma}_{h,B} - \gamma_{h,B}$, so $\mathbb{E}_t \xi_{h,B} = 0$. Then

$$\operatorname{Var}_t(\Delta m_{j^\star h}) = \eta^2 \operatorname{Var}_t(\xi_{h,B}) = \frac{\eta^2}{B^2} \sum_{b,c} \operatorname{Cov}_t(Y_{h,b,B}, Y_{h,c,B}) = \frac{\eta^2 \sigma_h^2}{B}. \tag{28}$$

The last equality requires conditionally iid sequence contributions with a law unchanged by B, or conditionally iid balancing groups of fixed size and law, with B divisible by that size. For fixed

groups, within-group dependence is absorbed into σ_h^2; token independence within sequences is unnecessary. The total coefficient includes task–auxiliary covariance. Global-batch load aggregation can change both the mean feedback (whose B subscript is suppressed in the main text) and covariance, so the covariance sum is the general expression. This margin-variance law needs no near-uniform assumption; excursions can receive subsequent reinforcement when net feedback is positive and the winner persists, but greater variance alone does not establish collapse.

Local entropy consequence. Assuming the same iid/fixed-group conditions for the full gradient contributions, write $\widehat{G}_{t,B} = G_t + \Xi_{t,B}$ with $\mathbb{E}_t \Xi_{t,B} = 0$ and $\mathbb{E}_{t,x}\|\Xi_{t,B}^\top x\|_2^2 = V_t/B$, including task–auxiliary covariance. For centered weights $W_t = W_{R,c}(t)$, define mean-step logits $\bar{\delta} = (W_t - \eta G_t)^\top x$ and noise $\epsilon = -\eta \Xi_{t,B}^\top x$. Taylor expansion at $\bar{\delta}$ gives

$$\begin{aligned} \mathbb{E}_{t,x} H(\mathrm{softmax}(\bar{\delta} + \epsilon)) &= \mathbb{E}_x H(\mathrm{softmax}(\bar{\delta})) - \frac{\eta^2 V_t}{2nB} + \mathcal{E}_{t,B}, \\ |\mathcal{E}_{t,B}| &\le C\, \mathbb{E}_{t,x}\left[\|\bar{\delta}\|_2 \|\epsilon\|_2^2 + \|\epsilon\|_2^3\right]. \end{aligned} \tag{29}$$

The linear term vanishes by conditional centering, and the entropy Hessian on centered logits is $-C_n/n + O(\|\bar{\delta}\|_2)$. Both mean-step and realized logits must remain in a fixed small neighborhood, with finite displayed moments. The leading noise-induced entropy decrease scales as η^2/B, relative to the same mean update. A batch-size comparison additionally needs a common mean update and control of the remainder; this is not an ordering of entire training trajectories.

D.4 Entropy Across Open-Source Checkpoints

Figure 5 shows layer-wise routing entropy on 64 WildBench (Lin et al., 2025) prompts. Most models' routing scores are distributed close to the uniform limit (black dashed lines). GPT-OSS-120B and Qwen3.5-35B-A3B both have normalized mean entropy of approximately 94.4%, despite markedly different weight similarities (~ 0 v.s. 0.505). Conversely, GLM-5.3-Flash and DeepSeek-V4-Flash-preview have relatively similar weight similarities (0.163 v.s. 0.180) but different normalized entropies (85.7% v.s. 49.3%), showing that similarity alone does not consistently describe cross-model routing concentration. We also note that most models show consistent routing behavior across layers, with only mild depth trends. The DeepSeek-V4-Flash family is a striking exception: per-token routing entropy approaches the uniform limit only in the first few MoE layers, then decreases markedly with depth, collapsing to a narrow effective expert pool in the deepest layers. It remains an open question whether it is a stable design characteristic of the family or a training artifact.

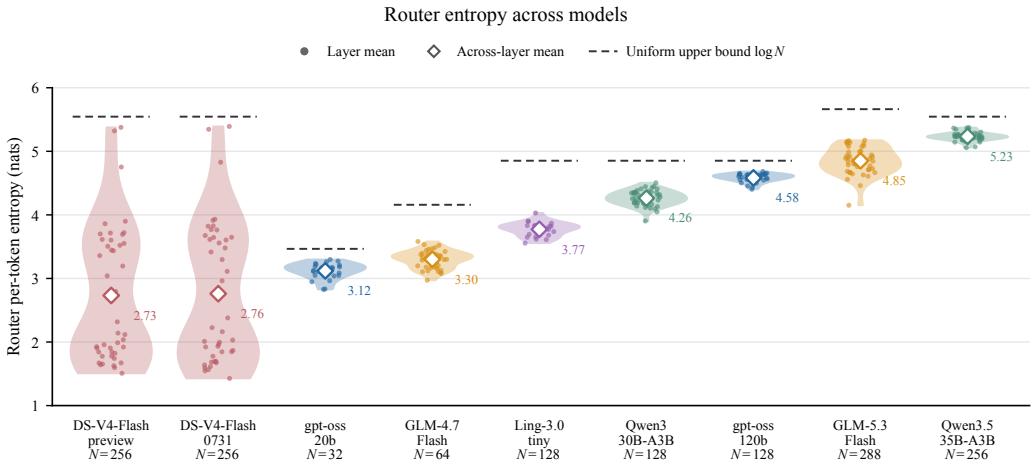


Figure 5: Layer-wise mean per-token routing entropy across checkpoints. We apply softmax to all expert logits and average token entropy within each layer on 64 English WildBench v1 queries during prefill. Dots denote layers, violins their distributions, and diamonds model means; n denotes the number of experts.

E Exact Low-Rank Spectral Computation and Monitoring Cost

Section 3.1 describes the exact small-core computation of the nonzero singular spectra of ΔF, ΔF_{lin}, and R. Here we benchmark its cost and numerical accuracy against full-matrix computation on a single Ascend 910B NPU. The compressed-core computation gives large speedups at realistic hidden sizes while preserving the spectrum to small relative error.

Size	Full eigvalsh (ms)	Core+eig. (ms)	Speedup	Rel. diff
100	0.55	3.80	0.14x	9.89×10^{-6}
200	39.23	4.54	8.64x	2.78×10^{-7}
500	66.04	3.86	17.12x	2.52×10^{-7}
1000	166.37	3.38	49.18x	2.50×10^{-7}
2000	1512.59	30.09	50.26x	2.29×10^{-7}
4000	3602.54	124.01	29.05x	2.26×10^{-7}

Table 3: Single Ascend 910B NPU timing for full-matrix eigendecomposition versus compressed-core computation of the same singular spectrum.