SPIKING, FAST AND SLOW: THE ANATOMY OF LOSS SPIKES IN LANGUAGE MODEL PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Loss spikes in language model pretraining can waste substantial compute and derail otherwise healthy runs. A primary driver of loss spikes is the growth and divergence of output logits in the final language modeling (LM) head. In this work, we divide these training instabilities into two types: fast spikes that abruptly increase loss before recovering, and slow terminal divergences that emerge only after prolonged training, which we call time-bomb spikes. We build on observations that these instabilities stem from fundamental properties of the softmax cross-entropy loss and its interaction with Adam's preconditioner. We observe that this interaction leads to LM head weight growth that may be harmless under exact arithmetic but can be destabilizing under finite-precision training and propose a simple optimization adjustment that ameliorates the issue. Furthermore, we show that various mitigation techniques from the literature (including our own) largely act on the shared underlying problem of logit growth by reducing quantization error in the LM head. Our work connects fast and time-bomb spikes and an array of prior methodological proposals through a common lens of analysis: controlling last-layer logit growth reduces finite-precision rounding error which is essential for stabilizing long pretraining runs.

1 INTRODUCTION

Pretraining Transformer-based language models at scale is notoriously difficult due to the instability of the objective and its propensity to diverge or "spike" during training (Chowdhery et al., 2023; Zhang et al., 2022; Zeng et al., 2023; OLMo et al., 2025). Wortsman et al. (2024) demonstrated that these large-scale training instabilities can be reproduced in small-scale Transformers by using higher learning rates as a proxy for natural spiking conditions. Using this insight, numerous works (Zhao et al., 2025; Stollenwerk et al., 2026; Su et al., 2025; Li et al., 2025) study large-scale training instabilities in small-scale transformers.

The particular type of instability we study in this work is closely related to divergence in the language model logits. Chowdhery et al. (2023) originally identified this correspondence and introduced z-loss as a mitigation strategy which has since become a standard safeguard in many training pipelines (Allal et al., 2025). However, while "band-aids" like z-loss suppress the symptoms of training instabilities involving logit-divergence, they do not address their root cause. We structure our research around a systematic decomposition of logit-divergence spikes into two categories: a type of fast spike that recovers quickly and a type of slow, unrecoverable spike that only occurs late in training.

Our experiments reveal that "fast spikes" are highly context-sensitive and attributable to only a small subset of outlier tokens with extreme loss values. Swapping even a single token hundreds of positions away from such high-loss tokens is often sufficient to collapse the loss by hundreds of nats, revealing that these fast spikes arise from a precise combination of input, model, and optimizer state rather than a broad training failure. However, when we attempt to leverage these observations and bypass fast spikes by skipping the inducing batches and reverting to a previous optimizer state, we find that the model still reliably spikes at some later point in training. This persistence suggests that such interventions treat only the surface symptoms of these spikes rather than their underlying cause.

We also study a late-emerging, unrecoverable type of "slow spike" associated with output-logit growth. We refer to these instabilities as "time-bomb" spikes because even though training initially appears healthy, a latent instability can still appear, ruining a long (expensive) training run. Suspecting that

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.

some more pervasive aspect of the update dynamics for the language modeling head itself must be to blame, we directly analyze the Adam optimizer's preconditioner, and find it to be the ultimate source of the slow drift in logit mean that accompanies divergence. By simply removing the preconditioner for the last layer and training it with only SGD plus momentum, we eliminate this drift and achieve more stable training across a range of model sizes and hyperparameter settings.

Finally, when evaluating our proposed mitigation against baselines from the literature, we uncover a clear correspondence between the characteristic logit growth of time-bomb spikes and increases in quantization error under low-precision arithmetic. After observing that increasing precision in the LM head serves as an effective stabilization method in its own right, and that all spike mitigations we consider appear to also control quantization error, we conclude that keeping features in the higher precision range of the chosen representation format may be critical. Our results suggest that the effectiveness of these disparate spike mitigations derives at least in part from this shared mechanism.

Our contributions are as follows: **1)** We identify logit growth and divergence as key correlates of both fast and slow loss spikes, and show that RMSNorm parametrization determines which types of spikes manifest. **2)** We show that fast spikes cannot be avoided by skipping the triggering examples or reverting optimizer state. These interventions only delay their occurrence indicating that individual batches expose, rather than cause, the underlying instability. **3)** We show that slow, time-bomb spikes emerge during overtraining, and are linked to logit growth stemming from output embedding anisotropy induced by the Adam optimizer's preconditioner in the language modeling head. We demonstrate that applying SGD-M to just those parameters fixes the issue. **4)** We develop a cohesive lens of interpretation for prior mitigation methods by showing that both our proposed method and those from prior work control quantization error in the LM head, leading to more stable training.

2 Preliminaries

In this work we investigate loss spikes stemming from output logit divergence in the language modeling head (Wortsman et al., 2024; Chowdhery et al., 2023). This type of training instability was originally identified during the training of PaLM (Chowdhery et al., 2023), where the authors proposed z-loss as a mitigation technique.

2.1 Language modeling head

Consider a vocabulary of size V. Let $\boldsymbol{y} \in \{0,1\}^V$ denote the one-hot encoded ground truth labels for a single example such that $\sum_{i=1}^{V} y_i = 1$. The model outputs logits $\boldsymbol{z} \in \mathbb{R}^V$, from which the next-token distribution $\boldsymbol{p}$ and cross-entropy objective are computed as

$$p_i = \text{softmax}(\boldsymbol{z})_i = \frac{\exp(z_i)}{\sum_{j=1}^{V} \exp(z_j)}, \quad i = 1, \ldots, V. \qquad \mathcal{L}_{\text{CE}} = -\sum_{k=1}^{V} y_k \log p_k$$

2.2 Baseline Mitigations for Logit Divergence

z-loss. Output logit divergence occurs when the absolute value of the output logits $\boldsymbol{z}$ grow excessively large. This can lead to both fast spikes, where loss increases quickly but eventually recovers, as well as slow loss divergences as studied in Wortsman et al. (2024). z-loss penalizes the logits for drifting too far from zero by introducing a regularization term based on $Z = \sum_{j=1}^{V} e^{z_j}$ such that the complete loss becomes $\mathcal{L} = \mathcal{L}_{CE} + \lambda \log^2 Z$.

μ-centering. More recently, Stollenwerk et al. (2026) introduced μ-centering to provably suppress output logit divergence. Unlike z-loss, which regularizes the loss objective, μ-centering is a structural intervention as it centers the output embeddings around zero. Let $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_V]^\intercal$ be the output embedding matrix. After each optimization step, the mean output embedding $\boldsymbol{\mu} = \frac{1}{V}\sum_{i=1}^{V} \boldsymbol{e}_i$ is subtracted from each individual output embedding e_i, creating new output embeddings $\boldsymbol{e}_i^* = \boldsymbol{e}_i - \boldsymbol{\mu}$. By enforcing $\sum_{i=1}^{V} \boldsymbol{e}_i^* = 0$, this procedure ensures the mean output logit remains zero, regardless of the hidden state magnitude.

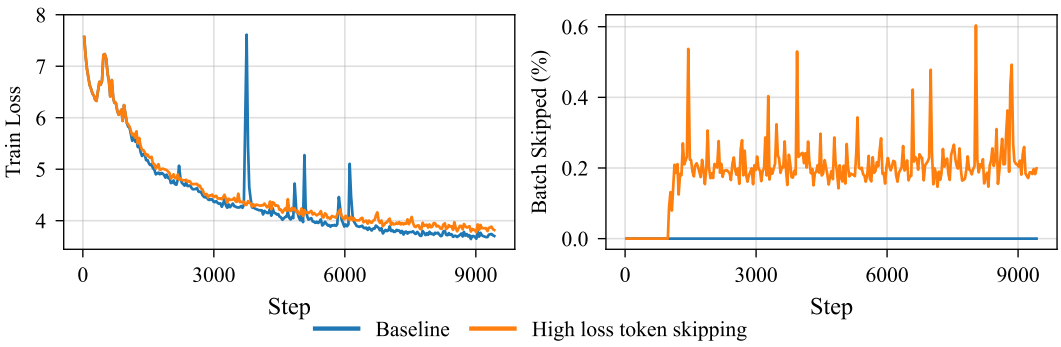


Figure 1: **Skipping a small fraction of high-loss tokens eliminates loss spikes but hurts convergence.** A 162M model trained at the Chinchilla-optimal token budget with peak learning rate 0.15. One run is the baseline; the other skips high-loss tokens identified via a rolling window threshold.

2.3 Shared Experimental Setup

To reproduce the loss spikes seen in larger models, we train 28M–1.4B parameter models at elevated learning rates following Wortsman et al. (2024) (fast and slow spike examples in Figures 4 and 8). Models are implemented in JAX (Bradbury et al., 2018) and Flax (Heek et al., 2024) and, unless otherwise noted, trained on up to 32 Google Cloud v6e or v4 TPUs in bfloat16 mixed precision. We use a modernized GPT-2 (Radford et al., 2019) architecture with rotary positional embeddings (Su et al., 2024), GELU activations (Hendrycks & Gimpel, 2023), RMSNorm (Zhang & Sennrich, 2019), untied embeddings, and no biases. We optimize with AdamW (Loshchilov & Hutter, 2019) ($\epsilon = 10^{-8}$, $\beta_1 = 0.9$, $\beta_2 = 0.95$) using linear warmup, cosine decay, and gradient clipping at a global norm of 1. Unless otherwise specified, weight decay is 10^{-4} and batch size is 262,144 tokens. For z-loss runs, $\lambda = 10^{-4}$ following Chowdhery et al. (2023).

All runs use QK-Norm (Dehghani et al., 2023), the recommended intervention against attention logit growth (Wortsman et al., 2024). This removes that source of spikes, so we can study instabilities from output logit divergence in isolation. All models are trained on FineWeb (Penedo et al., 2024) with the GPT-2 tokenizer and, unless otherwise noted, standard cross-entropy loss. Code will be released upon publication.

3 Fast Loss Spikes

We begin by systematically studying fast loss spikes through a series of observational and interventional experiments. We first examine how training sample characteristics and training metrics correlate with these instabilities, and test direct interventions that target the visible manifestations of a spike. Motivated by the persistence of spikes despite these interventions, and their apparent dependence on model normalization strategy, we conclude that some other underlying mechanism must be responsible.

Unless otherwise stated, these experiments use "parameter-less norms"—RMSNorm layers without learnable gain parameters. This design choice is used in several large language model suites (Groeneveld et al., 2024; Wang et al., 2026) and is also shared by prior work that studies or reports training instabilities during language model pretraining (Geiping et al., 2025; Filatov et al., 2026). However, we revisit this architectural choice later by directly comparing models trained with parameter-less norms and "parameterized norms" (with learnable gain vectors for each RMSNorm layer). We find that while models trained with parameterized norms exhibit fast spikes much less frequently, they still exhibit the slower type of time-bomb instability that we subsequently study in Section 4.

3.1 Fast Spikes Are Concentrated in Outlier Tokens

During a fast loss spike, we find that a small subset of tokens incur extremely large cross-entropy losses (100+) and therefore are responsible for the majority of the spike's apparent magnitude. To quantify these outliers, we compute a robust z-score using a rolling window of token losses from the preceding 256 steps (after an initial warmup period at the beginning of the run). This is similar in spirit to Karpathy (2024), but uses a robust z-score as we find token loss distributions to be heavy-tailed. As shown in Figure 1, masking as little as 0.2% of tokens is sufficient to eliminate major spikes. This masking, however, consistently hurts overall convergence in training loss.

Upon observing that skipping tokens can be effective at eliminating fast spikes, one might conclude that loss spikes stem from exposure to bad training examples and could be eliminated by filtering the data. We interrogate this hypothesis further in Section A.1 by identifying extreme outlier tokens and perturbing their contexts to gain a more mechanistic understanding of the role they play in causing a spike. While these token-level interventions are also effective at reducing the loss during a spike, the token types and context-sensitive perturbations required to achieve stability in this manner appear to be non-systematic and therefore do not admit a practical train-time strategy that one could deploy during a real training run.

3.2 Removing Triggering Batches Only Delays Fast Spikes

We also observe that fast loss spikes frequently coincide with sharp changes in the mean output logit, as shown in Appendix Figure 13. We therefore ask whether these events can be treated in the same way as the high-loss outlier tokens by skipping the *batch* that induces a loss spike. Following the batch-skipping procedure of Karpathy (2024), we find that skipping high-loss batches merely defers the instability rather than resolving it. As shown in Figure 2, after reverting to an earlier optimizer state, subsequent training again produces large changes in the output-logit mean and further loss spike events.

Together with the context perturbations in Section A.1, this suggests that individual batches do not trigger loss spikes, but rather an underlying instability in the model's parameters and optimizer states. Consequently, we hypothesize that data filtering of this kind actually targets the symptom rather than the underlying source of the instability.

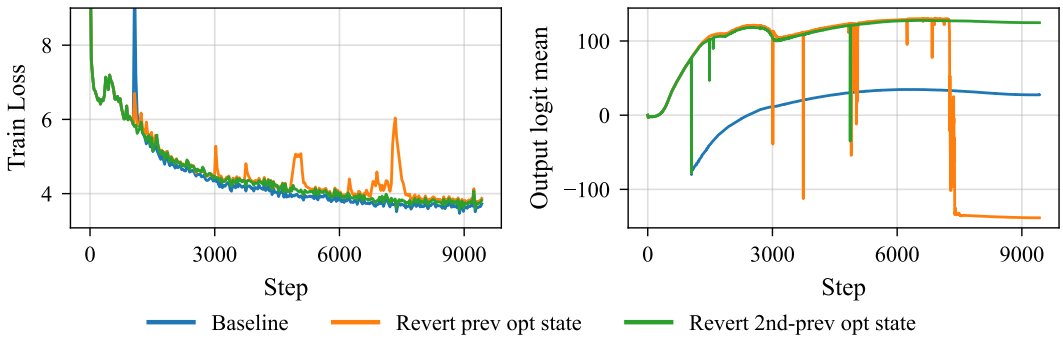


Figure 2: **Skipping high-loss batches defers rather than eliminates fast-spike instability.** Following Karpathy (2024), we skip any batch whose loss exceeds 7 standard deviations from a rolling 1000-step window and revert the optimizer state. Reverting to either the previous or second-to-previous optimizer state does not prevent subsequent fluctuations in output-logit mean characteristic of spike events.

3.3 Fast spikes depend strongly on normalization type.

The fast-spike experiments above use RMSNorm without learnable gain parameters. To test how strongly this architectural choice affects the phenomenon, we repeat a representative fast-spike configuration with learnable RMSNorm gains, holding all other hyperparameters and random seeds

fixed. Across five matched seed pairs, parameterized RMSNorms substantially reduce the fast loss spikes and the associated output logit mean and norm growth. Figure 3 shows a representative matched pair, with all five comparisons reported in Figure 12.

These results suggest that the abrupt fast-spike regime studied above is strongly affected by RMSNorm parameterization and is substantially less prominent with the learnable gains used in many modern language models (Yang et al., 2025; Shoeybi et al., 2019). Importantly, however, LM head instability does not disappear altogether. In the next section, we show that models with learnable RMSNorm gains can still develop a slower, delayed form of instability during long horizon training.

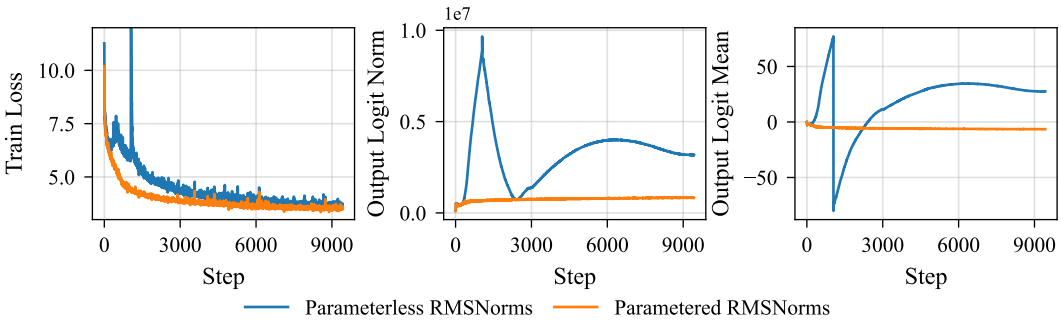


Figure 3: **Learnable gain parameters in normalization layers stabilize training and can prevent fast loss spikes.** Comparing a 162M spiking run with the same seed and configuration only differing in the inclusion of learnable RMSNorm parameters, the parameterized-norm configuration exhibits no spikes, output-logit swings, or large logit-norm growth.

4 Slow, Time-Bomb Loss Spikes

In the previous section analyzing fast spikes, all models were trained with "parameter-less norms"—RMSNorms without any learnable gain parameters. However, Figure 3 shows a hyperparameter and random seed matched run in which simply training a model with "parameterized norms" (RMSNorms *with* learnable gain parameters) also removes the sharp loss spike and substantially reduces output-logit mean drift and norm growth. This suggests that adopting parameterized norms as many modern language model architectures and codebases do (Yang et al., 2025; Shoeybi et al., 2019) greatly reduces the likelihood of observing fast spikes.

Therefore, in this section we turn our attention to another type of spike that persists even when using parameterized norms: slow, "time-bomb" spikes. Experiments in this section use RMSNorm layers with learnable gain parameters. All models use a batch size of 131,072 tokens and no weight decay; all other training hyperparameters follow the shared setup in Section 2. We use this configuration because preliminary experiments indicated that smaller batch sizes lead to earlier time-bomb onset, while weight decay delays their emergence (as also noted in Wortsman et al. (2024)). All models use bfloat16 mixed precision (BF16-mixed) by default unless otherwise noted.

4.1 Time-bomb loss spikes emerge during overtraining

To study time-bomb spikes, we "overtrain" 404M parameter models for 96B tokens across the same learning rate range as in Wortsman et al. (2024): $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 10^{-2}, 3 \times 10^{-2}, 10^{-1}, 3 \times 10^{-1}\}$. This corresponds to approximately 240 training tokens per parameter, or roughly $12\times$ the Chinchilla-optimal token budget. Unless otherwise specified, these experiments are conducted on four NVIDIA B200 GPUs. Figure 4 shows two runs that follow nearly identical loss trajectories through most of training, but where the higher-learning-rate run eventually stops improving and undergoes a sustained, monotonic increase in evaluation loss—a "time-bomb". Full loss and output-logit norm curves for the learning-rate sweep are shown in Appendix Figure 16. Longer training also leads to greater late-stage output-logit norm growth (Appendix Figure 17), suggesting that time-bomb spikes become more of an issue in the overtraining regime.

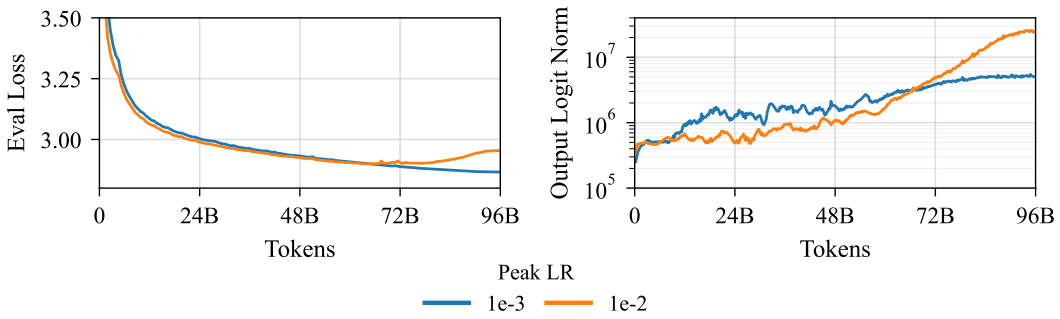


Figure 4: **Long horizon training can result in time-bomb instability.** Two 404M parameter models train similarly for most of pretraining, but the higher LR run exhibits delayed, unrecoverable loss increases accompanied by accelerating output-logit growth.

4.2 Adam's preconditioner induces LM head mean drift

Prior work has shown that the cross-entropy gradient of the LM head has a zero-sum structure along the vocabulary axis, while Adam's coordinate-wise preconditioning breaks this property (Biś et al., 2021; Stollenwerk & Stollenwerk, 2025). This can induce persistent drift in the mean output embedding and, by extension, in the mean output logits, directly linking the logit growth dynamics observed in the previous experiments to an interaction between the LM head gradients and Adam's preconditioner (Stollenwerk & Stollenwerk, 2025).

Let $\boldsymbol{E} \in \mathbb{R}^{V \times d}$ denote the output embedding matrix and $\boldsymbol{h} \in \mathbb{R}^d$ the incoming hidden activations, such that $\boldsymbol{z} = \boldsymbol{E}\boldsymbol{h}$. The cross-entropy gradient admits the following zero-sum property:

$$\frac{\partial \mathcal{L}_{CE}}{\partial \boldsymbol{E}} = (\boldsymbol{p} - \boldsymbol{y})\boldsymbol{h}^\intercal, \qquad \sum_{i=1}^{V} \frac{\partial \mathcal{L}_{CE}}{\partial E_{i,j}} = h_j \underbrace{\sum_{i=1}^{V} (p_i - y_i)}_{=0} = 0,$$

because softmax probabilities and one-hot labels both independently sum to one. Thus, any optimizer update that preserves this zero-sum structure leaves the mean output embedding over the vocabulary axis $\boldsymbol{\mu} = \frac{1}{V}\sum_i \boldsymbol{e}_i$ unchanged. Although Adam's first-moment estimate remains zero-sum, division by the coordinate-dependent second moment reweights vocabulary rows non-uniformly. Because these second moments vary strongly with token frequency (Stollenwerk & Stollenwerk, 2025), the resulting Adam update can have a non-zero vocabulary mean and induce drift in $\boldsymbol{\mu}$.

4.3 Mitigating Drift in the LM Head

The zero-sum structure motivates a simple optimizer-level intervention: train the LM head with SGD with momentum while retaining AdamW for the remainder of the model. Unlike Adam, SGD with momentum (SGD-M) does not apply coordinate-wise adaptive rescaling and therefore preserves the zero-sum structure of the LM head gradient. We compare this approach against AdamW, μ-centering, and z-loss.

Mitigation methods stabilize long horizon training. For LM head SGD-M, we tune the head learning rate separately and use increased momentum of 0.99. Figure 5 shows their behavior across the 404M-parameter learning rate sweep. All three methods improve stability relative to the AdamW baseline and substantially reduce output-logit growth. In particular, LM head SGD-M strongly constrains logit-norm growth while remaining competitive in final evaluation loss across the sweep. In Appendix Section A.3 we also compare these mitigations in the parameter-less norm, fast-spikes regime from Section 3 and find that they are similarly effective against this alternate class of instability.

For LM head SGD-M, the highest head learning rate in our grid achieves the best performance across all trunk learning rates. In order to measure whether head specific hyperparameter tuning gives SGD-M a disproportionate advantage, in Appendix Section A.6 and Figure 18 we explore the results

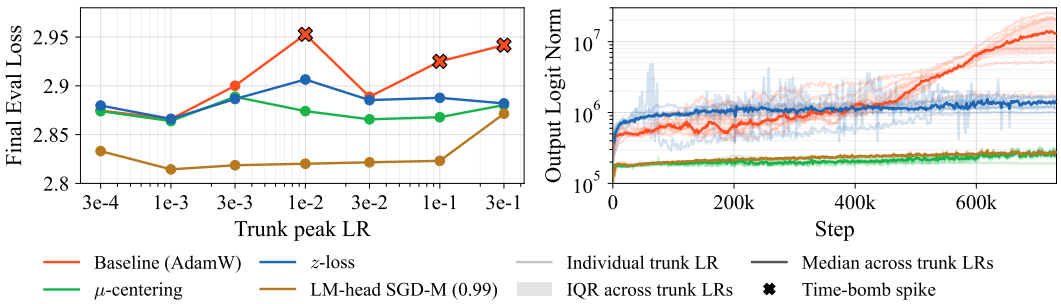


Figure 5: **SGD-M is competitive in both optimization quality and control of LM head growth.** Across trunk learning rates, μ-centering, z-loss, and LM head SGD-M improve stability relative to the AdamW baseline, while differing in both final evaluation loss (left) and the degree of LM head growth suppression (right). Across high learning rates, μ-centering and LM head SGD-M strongly suppress output-logit norm growth, whereas the baseline develops the largest late growth and corresponding time-bomb spikes (right). Faint curves show individual learning rates; solid curves and shaded regions show the median and interquartile range across selected learning rates.

of tuning the head LR and head specific β_1 value for AdamW. Tuning the AdamW head-specific learning rate alone provides only modest gains, but increasing the first-moment coefficient for the LM head to $\beta_1 = 0.99$ closes much of the optimization gap between AdamW and SGD-M. This is consistent with recent work showing that longer training horizons can favor longer optimizer memory and require retuning optimizer hyperparameters across the overtraining axis (Everett & Qiu, 2026). We also explore combining the higher-β_1 AdamW head with μ-centering and z-loss. However, despite their improved performance over the tuned AdamW at some learning rate settings, those combinations both also diverge at the largest trunk learning rate we tested.

4.4 Finite Precision's Role in Loss Spikes

Mean preservation alone cannot explain training stabilization. Although the analysis and experiments above connect Adam's preconditioner to drift in the vocabulary-mean output embedding, controlling this component alone cannot affect the cross-entropy computation in exact arithmetic.

To see why, let $\boldsymbol{E}' = \boldsymbol{E} - \mathbf{1}\boldsymbol{\mu}^\intercal$. For a fixed hidden state, this transformation shifts every logit by the same scalar and therefore leaves the softmax distribution unchanged (Stollenwerk et al., 2026). Because $\boldsymbol{p}$ is unchanged, the cross entropy gradient with respect to the LM head is consequently unchanged as well. Centering also leaves the gradient propagated into the hidden state unchanged:

$$\frac{\partial \mathcal{L}_{CE}}{\partial \boldsymbol{h}} = \boldsymbol{E}'^\intercal(\boldsymbol{p} - \boldsymbol{y}) = \boldsymbol{E}^\intercal(\boldsymbol{p} - \boldsymbol{y}) - \boldsymbol{\mu}\sum_{i=1}^{V}(p_i - y_i) = \boldsymbol{E}^\intercal(\boldsymbol{p} - \boldsymbol{y}).$$

Therefore, in settings without weight decay, and assuming exact computation, removing the vocabulary-mean component changes neither the model predictions nor the cross-entropy gradients through the LM head. Thus, in theory, growth in this mean component alone should not affect the model's predictions or optimization trajectory, nor should the centering performed by interventions like μ-centering meaningfully stabilize training.

However, in *finite* precision, a large logit mean implies an increase in the absolute numerical scale of the individual logit values, potentially reducing the precision with which the differences between logits that determine the predictive distribution can be represented. In our final set of experiments, we explore how the effect of limited precision in the LM head can help explain both the sensitivity to logit mean drift that real language models exhibit as well as the apparent effectiveness of mitigations like μ-centering and z-loss by accounting for changes in arithmetic representation error.

Reduced precision in the LM head accelerates timebombs. We vary LM-head precision across three settings: FP32, BF16 mixed, and BF16 true, ordered from highest to lowest numerical precision.

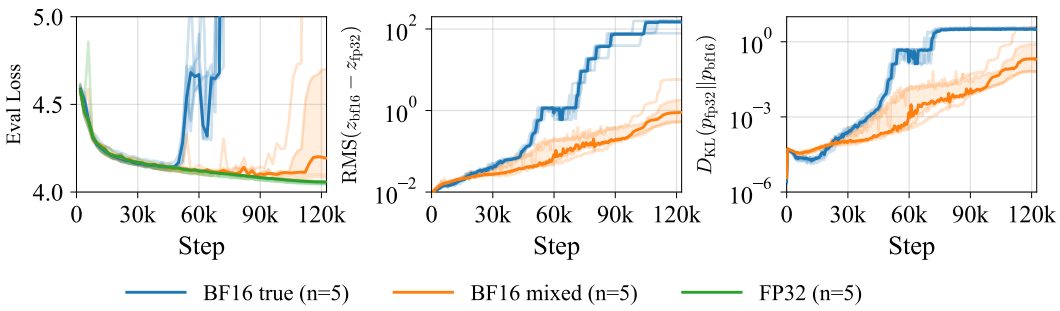


Figure 6: **Reduced precision in the LM head accelerates time-bomb instability and increases quantization error as logits grow.** For 28M parameter models trained at peak LR 0.1, BF16 true runs fail earliest, BF16 mixed runs deteriorate later, and FP32 remains stable across five seeds (left). The same ordering appears in the quantization error between BF16 native and FP32 reference logits (middle) and in the KL divergence between the corresponding predictive distributions (right), indicating that loss spikes coincide with increasingly large finite precision error. Solid lines show the median across seeds and faint lines show individual runs.

Each BF16 configuration uses the default BF16 general matrix multiply (GEMM) routine in the LM head computation. To directly measure numerical error in the LM head, at each logging step we recompute the LM head logits in FP32 from the same hidden states used by the native forward pass, yielding FP32 reference logits z_{fp32}. For these reference logits, the LM head GEMM is also performed in FP32. We compare these reference logits against the native BF16 logits z_{bf16}, measuring logit quantization error as the RMS distance, $\text{RMS}(z_{\text{bf16}} - z_{\text{fp32}})$, analogous to the commonly used reconstruction error in quantization literature (Frantar et al., 2023). We also measure the predictive-distribution quantization error as the KL divergence between the corresponding post-softmax distributions, $D_{\text{KL}}(p_{\text{fp32}} \| p_{\text{bf16}})$, following Lotfi et al. (2026).

Figure 6 shows that BF16 true runs develop substantial numerical error and time-bomb first, BF16 mixed runs fail a bit later, but FP32 runs remain stable over the training horizon. In both reduced-precision settings, increasing logit quantization error is accompanied by increasing predictive-distribution quantization error and an eventual loss spike. We present a deeper analysis of the precision-error correspondence in Section A.5.

All mitigation methods suppress quantization error. Finally, we test the hypothesis that different mitigation methods like z-loss and μ-centering may work in part because they suppress the growth in numerical errors associated with time-bomb instabilities. Figure 7 evaluates some of the methods from the previous section under BF16 mixed training. We observe that while the unmitigated baseline exhibits time-bomb loss spikes accompanied by a significant rise in quantization error in both logits and predictive distribution, in contrast, μ-centering, z-loss, and LM head SGD-M all prevent time-bomb failure and keep both error metrics bounded across five seeds. Thus, methods that stabilize LM head growth and prevent time-bomb spikes also suppress runaway growth in finite-precision error, suggesting an underlying mechanistic connection between these independently proposed approaches.

5 Related Work

Our work sits at the intersection of three research threads: empirical accounts of loss spikes, optimizer and architectural mitigations, and structural pathologies of the LM head.

Loss spikes recur across frontier-scale pretraining (Chowdhery et al., 2023; Zhang et al., 2022; Workshop et al., 2022; Zeng et al., 2023; DeepSeek-AI, 2026; Team et al., 2026). Examples of empirically-developed mitigations include rewinding to earlier checkpoints and skipping batches (Chowdhery et al., 2023; Zhang et al., 2022), lowering gradient-clipping thresholds (Pascanu et al., 2013), addressing fp16 issues at the embedding layer (Zeng et al., 2023), applying a StableEmbedding LayerNorm (Workshop et al., 2022; Dettmers et al., 2021), constraining attention logits via QK-norm (Henry et al., 2020; Dehghani et al., 2023) or QK-clip (Team et al., 2026), and periodically resetting

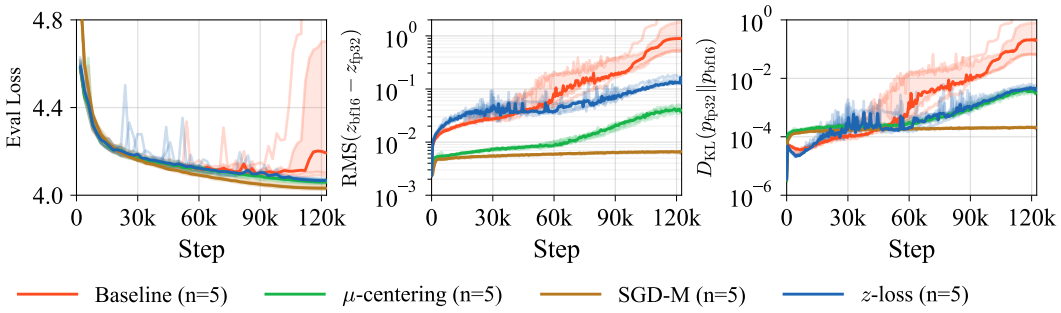


Figure 7: **Methods that prevent time-bomb failure also suppress quantization error.** For 28M-parameter models trained in BF16 mixed at peak LR 0.1, the unmitigated baseline develops time-bomb spikes (left) together with orders-of-magnitude growth in logit quantization error (middle) and predictive distribution quantization error (right). All mitigation methods prevent the time bomb and keep both numerical-error metrics bounded across five seeds. Solid lines show medians and faint lines show individual runs. The same effect is more pronounced under BF16-true training; the unmitigated baseline fails catastrophically after significant growth in numerical error (Figure 19).

Adam's momentum buffers (Huang et al., 2025). Relevant to our findings, replaying the same batches from different checkpoints failed to reproduce PaLM's spikes (Chowdhery et al., 2023), ruling out pure-data explanations, and Wortsman et al. (2024) reproduced similar instabilities at small scales via elevated learning rates, enabling systematic studies like ours.

Two output logit-targeting mitigations dominate. z-loss (Chowdhery et al., 2023) adds an auxiliary penalty on $\log Z$ and is now standard in frontier pretraining (Allal et al., 2025). μ-centering (Stollenwerk et al., 2026) constrains the row mean of the output embedding to zero at every step (Section 2). Stollenwerk & Stollenwerk (2025) more directly attribute embedding anisotropy to the interaction of long-tailed token frequencies with Adam's per-coordinate v_t, and propose sharing Adam's second moment across the vocabulary axis as a remedy.

A parallel line of theoretical work documents structural weaknesses of the LM head: Yang et al. (2017) prove the softmax bottleneck (the rank-d projection cannot express the distributions natural language requires when $V \gg d$), and Godey & Artzi (2026) extend this to the optimizer side, finding the output projection absorbs 95–99% of the backpropagated gradient norm in standard models.

6 Conclusion

We study two types of loss spikes in language model pretraining that arise from output logit divergence: fast, transient spikes and slow, time-bomb spikes. Our experiments show that fast spikes are preceded by drift in the mean output logit and are tightly coupled to a small subset of outlier tokens whose loss can be collapsed by perturbing only their local context. However, interventions targeting these symptoms, such as skipping high-loss tokens or optimizer-state rollbacks, only defer these types of instabiliites. Our analysis of the long-term logit drift characteristic of spikes originating from the LM head can be traced to the interaction between Adam's preconditioner and the zero-sum structure of the cross-entropy gradient along the vocabulary axis. Because Adam's $1/\sqrt{v}$ rescaling varies by orders of magnitude across rows, applying it to a zero-sum gradient yields a non-zero sum, causing the mean output embedding to drift. Existing mitigations such as μ-centering restore the zero-sum property post-hoc while SGD-M prevents it in the first place.

Finally, we derive an important connection between the symptoms of loss spikes and the mitigations proposed in prior work. Theoretically harmless logit drift can turn into full blown divergence by pushing logits outside of the nominal range of low-precision arithmetic. By showing that training just the LM head in high precision is sufficient to remove all observed instabilities without any other mitigation, we conclude that the mitigations proposed in both prior work, as well as our own, may largely act via a shared mechanism: controlling growth in arithmetic error by keeping logit values within a higher-precision regime.

AI USE STATEMENT

In this work, we used generative AI tools for refining hypotheses, providing feedback on experiments, interpreting results, supporting qualitative and thematic data analysis, and implementing methods. We have not used generative AI tools for cleaning and formatting datasets, and generating synthetic data sets, helping develop theoretical models, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, are not applicable to this work.

Additionally, we used generative AI tools for creating scientific figures, suggesting experimental parameters, creating and editing software code, discovering research topics and identifying gaps, editing a research paper to improve readability, and brainstorming. We have reviewed all AI-assisted work. We have verified that LLM-retrieved papers were relevant and correctly cited. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Loubna Ben Allal, Lewis Tunstall, Nouamane Tazi, Elie Bakouch, Ed Beeching, Carlos Miguel Patiño, Clémentine Fourrier, Thibaud Frere, Anton Lozhkov, Colin Raffel, Leandro von Werra, and Thomas Wolf. The smol training playbook: The secrets to building world-class llms, 2025.

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. Too much in common: Shifting of embeddings in transformer language models and its implications. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5117–5130, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.403. URL https://aclanthology.org/2021.naacl-main.403/.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

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, Sashank 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(1), January 2023. ISSN 1532-4435.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

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.), *Proceedings of the 40th International Conference on Machine*

Learning, volume 202 of *Proceedings of Machine Learning Research*, pp. 7480–7512. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/dehghani23a.html.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*, 2021.

Katie Everett and Shikai Qiu. Optimizer memory schedules for outscaling the overtraining axis, 2026. URL https://arxiv.org/abs/2609.04577.

Oleg Filatov, Jiangtao Wang, Jan Ebert, and Stefan Kesselheim. Optimal scaling needs optimal norm, 2026. URL https://arxiv.org/abs/2510.03871.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Jonas Geiping, Sean Michael McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=S3GhJooWIC.

Nathan Godey and Yoav Artzi. Lost in backpropagation: The lm head is a gradient bottleneck, 2026. URL https://arxiv.org/abs/2603.10145.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15789–15809, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.841. URL https://aclanthology.org/2024.acl-long.841/.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024. URL http://github.com/google/flax.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. URL https://arxiv.org/abs/1606.08415.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Tianjin Huang, Ziquan Zhu, Gaojie Jin, Lu Liu, Zhangyang Wang, and Shiwei Liu. SPAM: Spike-aware adam with momentum reset for stable LLM training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=L9eBxTCpQG.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Andrej Karpathy. Cool! for the spike i'd try e.g. '-sl 7 -sg 7' to keep instability in check earlier in the training. (will skip update if loss/gradnorm > 7 sigma outlier is detected). X (formerly Twitter), July 2024. URL https://x.com/karpathy/status/1812917107379872145.

Yunshui Li, Yiyuan Ma, Shen Yan, Chaoyi Zhang, Jing Liu, Jianqiao Lu, Ziwen Xu, Mengzhao Chen, Minrui Wang, Shiyi Zhan, Jin Ma, Xunhao Lai, Yao Luo, Xingyan Bin, Hongbin Ren, Mingji Han, Wenhao Hao, Bairen Yi, LingJun Liu, Bole Ma, Xiaoying Jia, zhou Xun, liang xiang, and Yonghui Wu. Model merging in pre-training of large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=HW55AwGEC8.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Sanae Lotfi, Polina Kirichenko, Steven Li, and Zechun Liu. Quantized reasoning models think they need to think longer, but they do not, 2026. URL https://arxiv.org/abs/2606.00206.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious, 2025. URL https://arxiv.org/abs/2501.00656.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks, 2013. URL https://arxiv.org/abs/1211.5063.

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

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

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Felix Stollenwerk and Tobias Stollenwerk. Better embeddings with coupled Adam. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27219–27236, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1321. URL https://aclanthology.org/2025.acl-long.1321/.

Felix Stollenwerk, Anna Lokrantz, and Niclas Hertzberg. Output embedding centering for stable llm pretraining, 2026. URL https://arxiv.org/abs/2601.02031.

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

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

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, 2026. URL https://arxiv.org/abs/2507.20534.

Guan Wang, Changling Liu, Chenyu Wang, Cai Zhou, Yuhao Sun, Yifei Wu, Shuai Zhen, Luca Scimeca, and Yasin Abbasi Yadkori. Hrm-text: Efficient pretraining beyond scaling, 2026. URL https://arxiv.org/abs/2605.20613.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

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*, 2024. URL https://openreview.net/forum?id=d8w0pmvXbZ.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W Cohen. Breaking the softmax bottleneck: A high-rank rnn language model. *arXiv preprint arXiv:1711.03953*, 2017.

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*, 2023. URL https://openreview.net/forum?id=-Aw0rrrPUF.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems 32*, Vancouver, Canada, 2019. URL https://openreview.net/references/pdf?id=S1qBAf6rr.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham M. Kakade. Deconstructing what makes a good optimizer for autoregressive language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zfeso8ceqr.

A Appendix

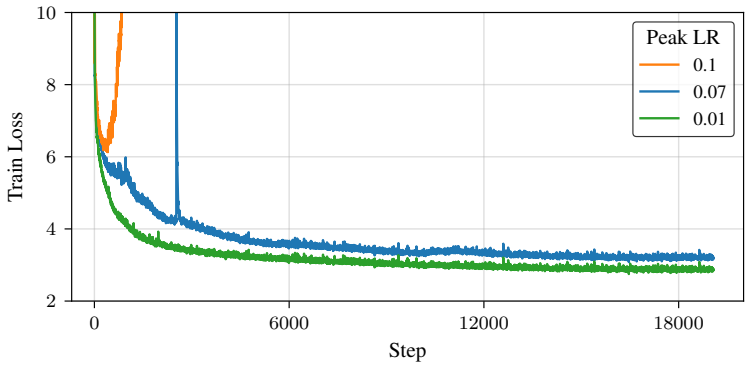


Figure 8: **Learning rates can control the type of training instability elicited**. Example training runs at 1.4B scale trained for 10B tokens. We focus on the learning rates similar to the middle one in order to elicit fast loss spikes to analyze.

A.1 Swapping out bad tokens.

We rank validation-set token predictions by cross-entropy loss at the peak of the spike, aggregating across all batches, sequences, and positions; we refer to the highest-loss prediction as the **Rank-1 event**. For this prediction ($\mathcal{L} = 610.0$), the model is given the context "`...the commencement of the 50th Commemoration of that War.The Command`" and asked to predict the next token `'ery'` (completing Command**ery**"). As shown in Table 1, this is driven by a logit explosion where the top-predicted token receives a logit value of 782.05. Replacing the most recent context token (`Command'`) with a comma (id 11), while keeping the target `ery'` fixed, reduces the loss to 12.73, a delta of -594.97 from the original. Moreover, the maximum logit magnitude contracts dramatically.

Figure 9 contrasts the full-vocabulary logit distributions ($n = 50{,}260$) at the Rank-1 loss event before and after the swap. The "before swap" distribution has a standard deviation of 121.87, and even low-probability tokens are assigned massive logit values. Conversely, the "after swap" distribution shows a substantially narrower logit distribution. The mean shifts from -57.46 to $+77.46$, and the spread narrows to 6.87, which is characteristic of the model's behavior during non-spiking steps.

These patterns can occur for every sequence position in the context and for a variety of swap tokens, including the single space `' '` and empty space `''` tokens. We conclude that spike events are highly sensitive to context: small perturbations to the tokens preceding the target token are often sufficient to recover normal loss values, and spike events themselves correspond to the model placing nearly all probability mass on a single wrong token.

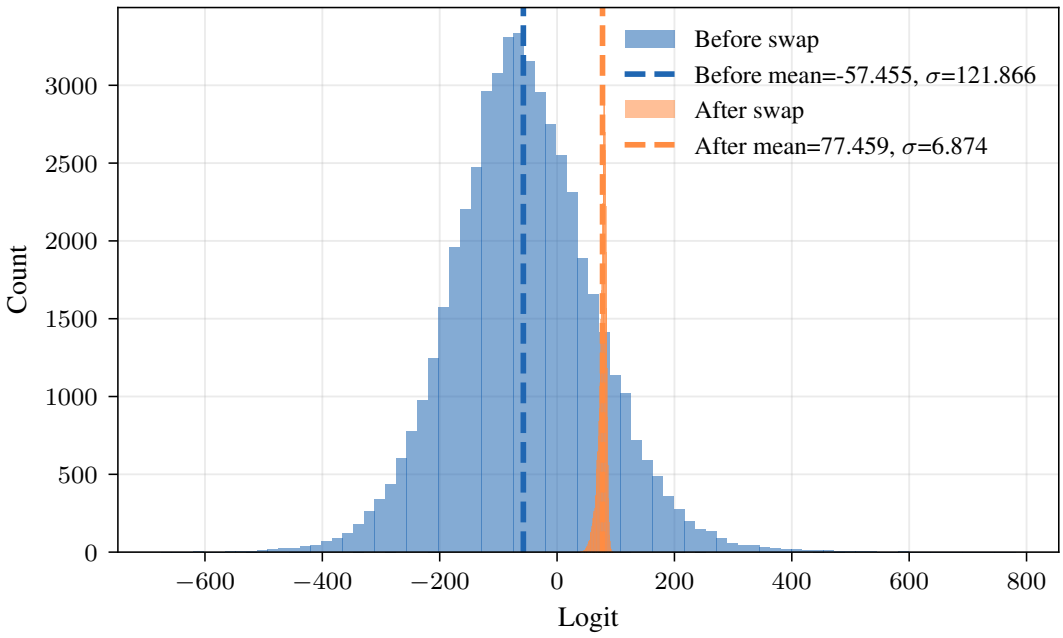


Figure 9: **A single-token context change collapses the logit distribution at a spike.** Full-vocabulary logits ($V = 50{,}260$) at the Rank-1 spike event before (blue) and after (orange) replacing the final context token (`Command` → `,`), with the target token `ery` held fixed. The standard deviation contracts $17\times$ (from $\sigma = 121.87$ to $\sigma = 6.87$) and the mean shifts from -57.46 to $+77.46$, recovering the well-behaved range characteristic of non-spiking steps.

A.2 Detailed top-5 logit shift for the Rank-1 spike event.

	Original Context (Spiking)		Swapped Context (Recovered)	
Rank	**Token**	**Logit**	**Token**	**Logit**
1	`'ible'`	782.05	`' the'`	95.01
2	`'ments'`	669.86	`' in'`	94.12
3	`'ured'`	640.67	`' however'`	93.80
4	`'ive'`	603.56	`' until'`	92.99
5	`'ivism'`	595.26	`' and'`	92.96
Target	**`'ery'`**	—	**`'ery'`**	—
Loss		**610.00**		**12.73**

Table 1: **A single-token context swap collapses the spike.** Top-5 predicted tokens before and after replacing the final context token (`'Command'`) with the comma token (`','`, id 11). The pre-swap distribution places extreme mass on the wrong token `'ible'` (logit 782.05); after the swap, the distribution flattens and cross-entropy loss drops from 610.00 to 12.73.

A.3 Fast spikes mitigation results

Surgically restoring zero-sum for the LM head. Several corrections are possible: μ-centering (Stollenwerk et al., 2026) restores the zero-sum property by column-centering the post-Adam update, but requires communication across the vocabulary axis and couples the update dynamics of rare and frequent tokens. Replacing Adam with SGD-M on the LM head avoids inducing the violation in the first place, removing the need for any post-hoc correction. We additionally evaluate a purely local variant, *row-centering*, which subtracts each token's row mean across the hidden dimension ($M^t_{i,j} \leftarrow M^t_{i,j} - (1/d)\sum_{k=1}^{d} M^t_{i,k}$); results are reported in Table 3.

Table 2: **Learning-rate sweep across the spike-prone regime ($\eta \in [10^{-1}, 3{\times}10^{-1}]$) for 162M and 404M models.** Final evaluation loss (↓ lower is better) is reported. † marks runs whose loss spiked during training and did not recover.

	Peak learning rate η (final eval loss ↓)						
Method	10^{-1}	$1.2{\cdot}10^{-1}$	$1.45{\cdot}10^{-1}$	$1.75{\cdot}10^{-1}$	$2.1{\cdot}10^{-1}$	$2.5{\cdot}10^{-1}$	$3{\cdot}10^{-1}$
162M							
AdamW	3.642	3.679	3.733	3.733	12.552†	23.223†	10.845†
μ-centering	3.637	3.656	3.674	3.701	4.094†	6.063†	5.950†
z-loss	3.636	3.655	3.690	3.718	5.594†	5.964†	7.495†
row-centering	**3.617**	**3.631**	**3.633**	3.718	3.697	3.793	3.863
SGD-M, β=0.9	3.926	3.891	3.869	3.849	3.821	3.803	3.789
SGD-M, β=0.99	3.683	3.674	3.659	**3.649**	**3.642**	**3.666**	**3.736**
404M							
AdamW	3.295	3.323	3.248	11.125†	11.006†	16.731†	28.241†
μ-centering	3.172	3.197	3.221	3.231	5.144†	6.088†	7.115†
z-loss	**3.161**	3.183	3.213	3.249	3.289	7.266†	6.588†
row-centering	3.257	**3.166**	3.550	4.930†	12.106†	3.310	3.452
SGD-M, β=0.9	3.335	3.319	3.304	3.280	3.268	3.255	3.244
SGD-M, β=0.99	3.186	3.183	**3.180**	**3.173**	**3.169**	**3.167**	**3.167**

Learning-rate sweep. We compare the mitigation methods across learning rates $\eta \in [10^{-1}, 3 \times 10^{-1}]$, spanning the regime in which fast spikes emerge. As shown in Figure 11, the methods differ substantially in both the range over which they suppress spikes and their final evaluation loss. μ-centering and z-loss prevent fast spikes over much of the sweep but become unstable at some higher learning rates, while LM head SGD-M with $\beta = 0.99$ remains stable across the full range tested. At higher learning rates, SGD-M frequently attains the lowest final evaluation loss while remaining stable across the full range tested. Full numerical results are reported in Section A.3.

Spike frequency across seeds. To compare stability across runs, we follow OLMo et al. (2025) and define the *spike score* as the percentage of training steps whose loss exceeds seven standard deviations from a trailing 1000-step window. Table 3 reports both spike score and final evaluation loss across seeds and model scales, allowing us to distinguish suppression of visible spikes from differences in final evaluation loss. The mitigation methods reduce fast-spike incidence to differing degrees, with LM head SGD-M exhibiting low spike scores across the settings considered.

We additionally repeat the comparison at 1.4B scale. As shown in Figure 10, the methods again differ in their behavior at larger scale; in this experiment, LM head SGD-M reaches a lower training loss than AdamW, μ-centering, and z-loss. We report the corresponding results with a Muon (Jordan et al., 2024) backbone in the same figure.

Table 3: **Tuned SGD-M on the LM head suppresses spikes and remains competitive on evaluation loss.** Final evaluation loss and training-loss spike score (defined in Section A.3) at peak learning rate $\eta = 0.15$ for 162M and 404M models. Values are mean ± standard error across 5 seeds; bold marks the best result per column.

	162M		**404M**	
Method	Eval Loss ↓	Spike Score ↓	Eval Loss ↓	Spike Score ↓
Baseline (AdamW)	3.702 ± 0.014	0.088 ± 0.029	3.916 ± 0.278	0.085 ± 0.029
z-loss	3.644 ± 0.006	0.064 ± 0.027	3.214 ± 0.008	0.106 ± 0.040
μ-centering	3.646 ± 0.009	0.066 ± 0.043	3.208 ± 0.004	0.064 ± 0.002
Row-centering	$\mathbf{3.621 \pm 0.016}$	0.038 ± 0.007	4.858 ± 1.438	0.056 ± 0.011
SGD-M, $\beta = 0.9$	3.855 ± 0.003	0.038 ± 0.004	3.301 ± 0.003	0.060 ± 0.002
SGD-M, $\beta = 0.99$	3.644 ± 0.005	0.038 ± 0.004	$\mathbf{3.178 \pm 0.002}$	0.069 ± 0.003

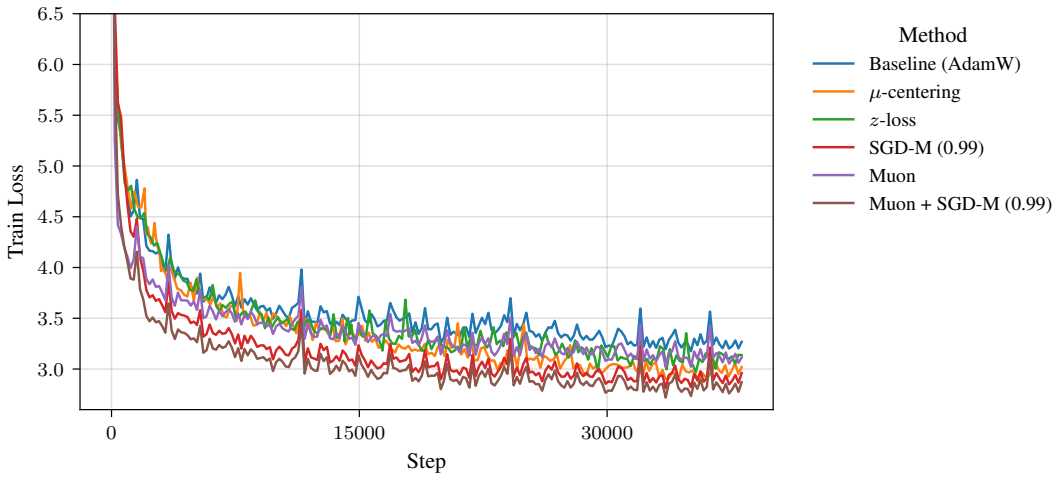


Figure 10: **SGD-M on the LM head improves on every baseline mitigation at 1.4B scale, and pairs effectively with Muon on the backbone.** Training loss curves for 1.4B models at peak learning rate $\eta = 0.07$. Replacing AdamW with SGD-M ($\beta = 0.99$) on the LM head reaches a lower final loss than AdamW, μ-centering, and z-loss; the combination of a Muon backbone with SGD-M on the LM head reaches the lowest loss of any method shown.

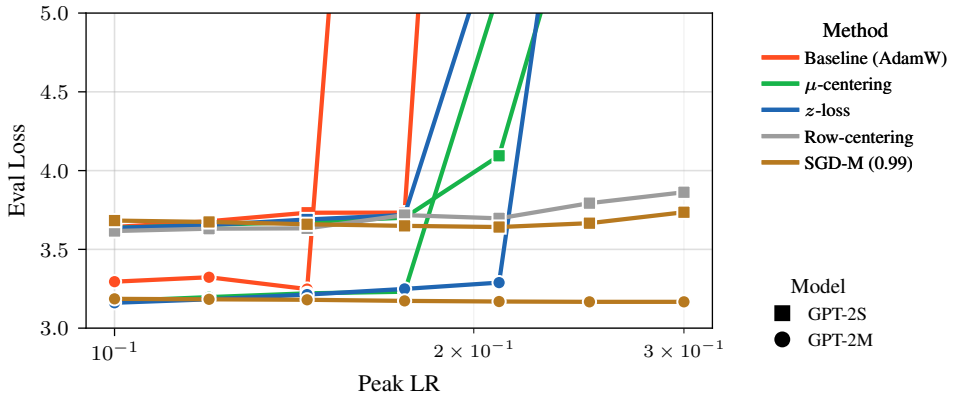


Figure 11: **SGD-M on the LM head is the only intervention stable across the spike-prone LR regime.** Final evaluation loss vs peak learning rate for 162M and 404M models. Interventions besides SGD-M on the LM head diverges at some $\eta \in [10^{-1}, 3 \times 10^{-1}]$, while SGD-M on the LM head remains stable throughout and achieves the lowest loss at the high end. Row-centering omitted for GPT2-M due to inconsistent training divergence across LRs.

A.4 Output-logit mean dynamics during fast spikes

Fast spikes coincide with sharp swings in the mean output logit. As shown in Figure 13, these changes occur alongside the onset of a spike and are absent in an otherwise matched non-spiking run differing only in random seed.

A.5 A potential numerical explanation for error growth with logit mean

In Section 4.4 we connected time-bomb instabilities coinciding with logit-growth throughout the run to effects of limited precision computation. Here we sketch a numerical explanation for this behavior based on BF16's relative precision, and show that normalized logit error grows strongly as the logit mean becomes large relative to the variation across logits. The full learning-rate sweep failure rates are also provided in Figure 15.

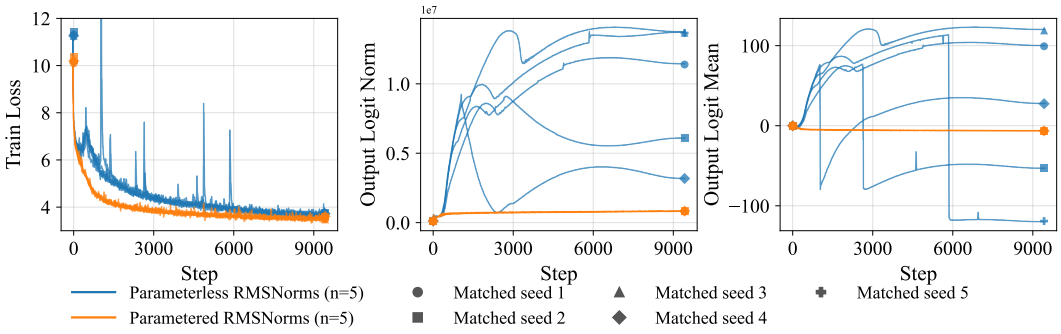


Figure 12: **Parameterized RMSNorms effectively mitigate fast loss spikes.** The five parameterized runs are highly consistent and overlap closely, indicating low seed-to-seed variability relative to the parameter-less runs.

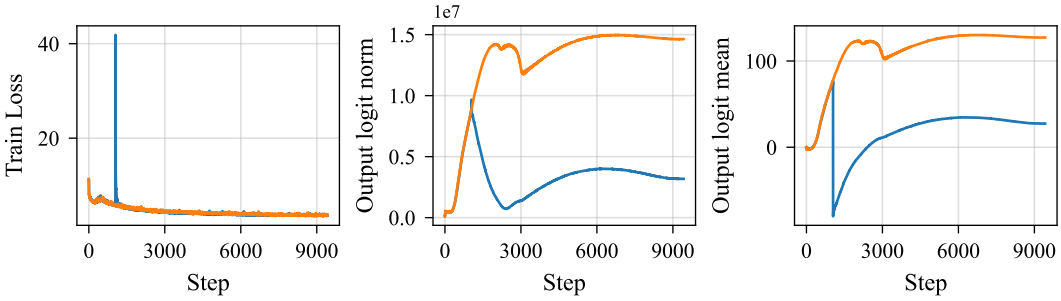


Figure 13: **Fast loss spikes coincide with sharp swings in the output-logit mean.** A spiking 162M run compared with a non-spiking run using the same configuration and differing only in random seed.

For normal BF16 values $x \in [2^e, 2^{e+1})$, adjacent representable values are separated by

$$\mathrm{ULP}(x) = 2^{e-7}.$$

Thus, the absolute spacing between representable values increases with their magnitude. A large logit mean can therefore increase the numerical scale of the logits without increasing the differences between logits that determine the softmax distribution. This motivates the ratio

$$r = \frac{|\mu_z|}{\sigma_z},$$

which measures the magnitude of the logit mean relative to the variation across logits.

As shown in Figure 14, the normalized numerical error

$$\frac{\mathrm{RMS}(z_{\mathrm{native}} - z_{\mathrm{fp32}})}{\mathrm{std}(z_{\mathrm{fp32}})}$$

grows strongly as r increases across both BF16 precision settings and learning rates. This relationship is consistent with BF16's fixed relative precision: as the logit mean grows relative to the variation across logits, the available numerical resolution becomes increasingly coarse relative to that variation.

A.6 LM HEAD HYPERPARAMETER TUNING

Because LM head SGD-M uses a separately tuned learning rate and momentum coefficient, we additionally tune LM-head-specific AdamW hyperparameters. We consider tuning the LM head learning rate independently of the trunk learning rate, increasing the LM-head first-moment coefficient from the default $\beta_1 = 0.9$ to $\beta_1 = 0.99$, and combining these changes with μ-centering and z-loss.

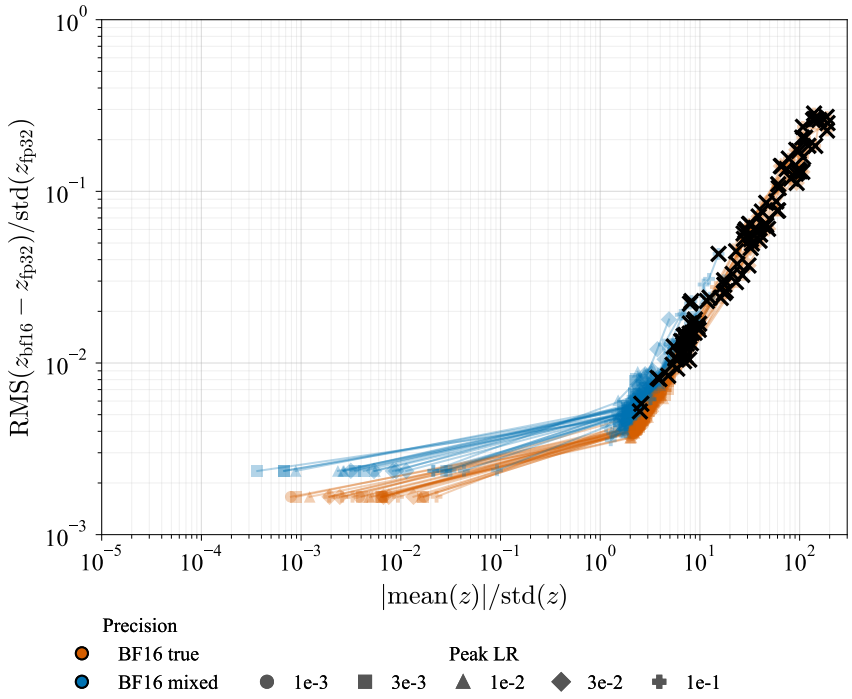


Figure 14: **Normalized logit error grows as the logit mean becomes large relative to variation across logits.** Across BF16 true and BF16 mixed baseline runs, the normalized RMS error between native BF16 logits and FP32 reference logits, $\mathrm{RMS}(z_{\mathrm{bf16}} - z_{\mathrm{fp32}})/\mathrm{std}(z_{\mathrm{fp32}})$, increases with the relative logit mean, $|\mathrm{mean}(z)|/\mathrm{std}(z)$. Colors indicate precision, markers indicate peak learning rate, and lines trace individual training trajectories.

	10^{-3}	3×10^{-3}	10^{-2}	3×10^{-2}	10^{-1}
FP32	–	–	–	–	0/5
BF16 mixed	–	0/5	0/5	2/5	5/5
BF16 true	0/5	4/5	5/5	5/5	5/5

Figure 15: **Reduced precision shifts the time-bomb stability boundary.** Fraction of seeds exhibiting a time-bomb instability across peak learning rates and numerical precision settings. As precision decreases from FP32 to BF16 mixed to BF16 true, instability emerges at progressively smaller learning rates. A time bomb is defined as three consecutive post-midpoint evaluations at least 0.01 above the running-best evaluation loss.

As shown in Figure 18, tuning the LM-head learning rate alone provides only modest improvement, whereas increasing β_1 produces a substantially larger gain in final evaluation loss.

These runs were accumulated across three hardware configurations: $4\times$ NVIDIA B200, $8\times$ NVIDIA H100, and $8\times$ NVIDIA A100 GPUs.

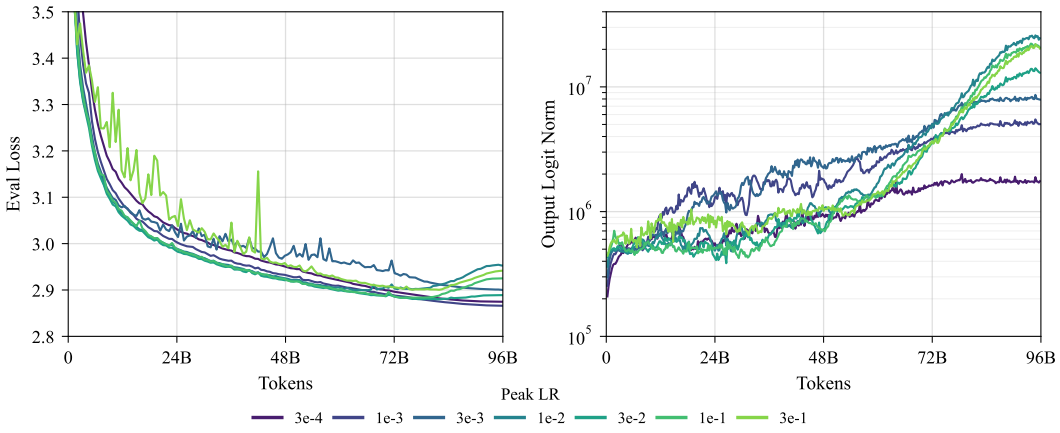


Figure 16: **Time-bomb instability emerges across the long-horizon learning-rate sweep alongside accelerating output-logit growth.** Evaluation loss (left) and output-logit norm (right) for 404M-parameter models trained for 96B tokens across peak learning rates from 3×10^{-4} to 3×10^{-1}. Runs train similarly for much of pretraining, while higher-learning-rate settings exhibit substantially greater late-stage logit-norm growth and, in several cases, deterioration in evaluation loss.

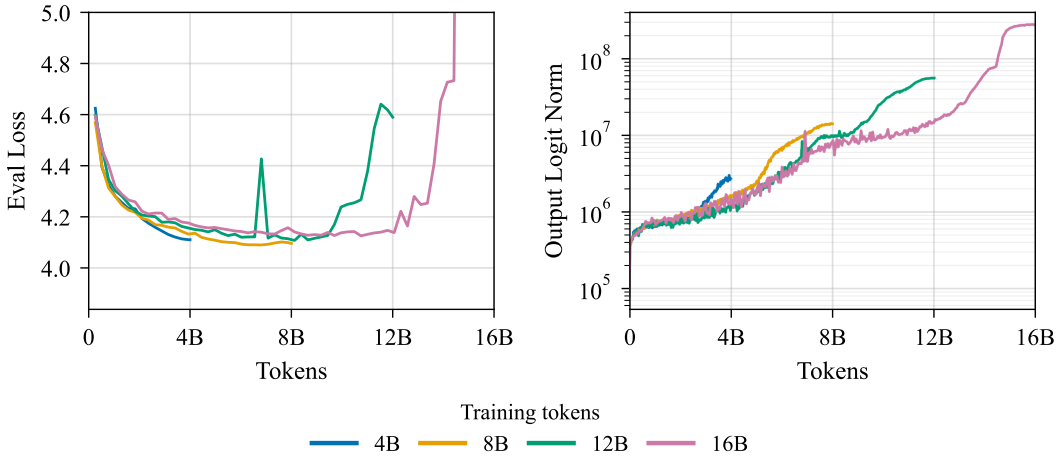


Figure 17: **Longer training horizons are associated with increasingly severe late-stage LM-head growth.** Evaluation loss (left) and output-logit norm (right) for 28M parameter models with parameterized RMSNorm, and peak LR of 0.1 trained with horizons of 4B, 8B, 12B, and 16B tokens. Extending training exposes progressively larger output-logit norm growth and late-stage loss deterioration, consistent with time-bomb instability becoming more prominent during prolonged training.

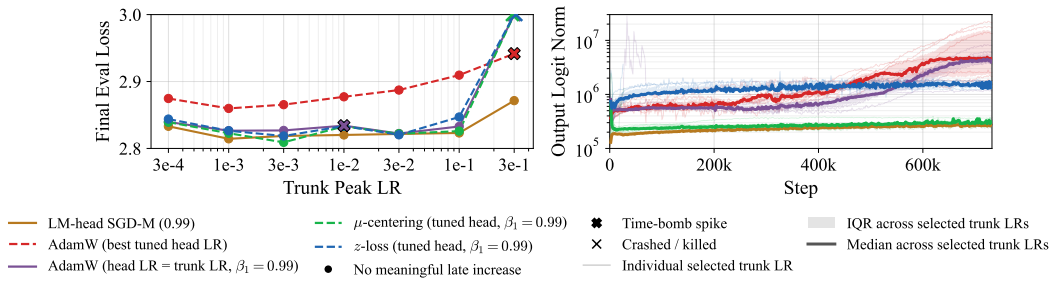


Figure 18: **Increasing LM-head β_1 is more effective than tuning the LM-head learning rate alone. Left:** Final evaluation loss across trunk learning rates. Tuning the AdamW LM-head learning rate provides only modest gains, whereas setting the LM-head first-moment coefficient to $\beta_1 = 0.99$ closes much of the gap to LM-head SGD-M. Combining $\beta_1 = 0.99$ with head-LR tuning and either μ-centering or z-loss improves performance further, with μ-centering achieving the lowest overall loss across the sweep. At the largest trunk learning rate, however, these AdamW-based mitigations still exhibit time-bomb instability or divergence, while LM-head SGD-M remains stable. **Right:** Output-logit norm trajectories for representative trunk learning rates, showing the differing degree to which each method constrains LM-head growth.

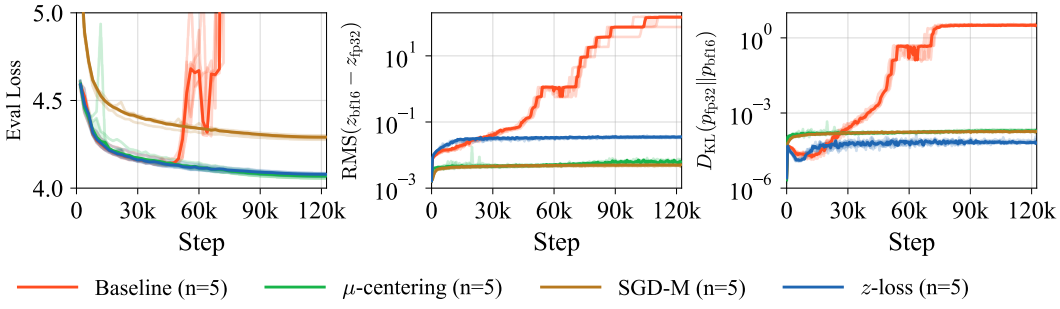


Figure 19: **Methods that prevent time-bomb failure also suppress quantization error.** For 28M-parameter models trained in BF16-true at peak learning rate 0.1, the unmitigated baseline develops severe unrecoverable loss spikes (left) together with orders-of-magnitude growth in logit quantization error (middle) and predictive distribution quantization error (right). μ-centering, z-loss, and LM head SGD-M prevent the time bomb and keep both numerical-error metrics bounded across five seeds. Solid lines show medians and faint lines show individual runs.