Adaptive Preconditioners Trigger Loss Spikes in Adam

Zhiwei Bai[1, †], **Zhangchen Zhou**[1, †], **Jiajie Zhao**[1], **Xiaolong Li**[1], **Zhiyu Li**[3,4], **Feiyu Xiong**[3,4], **Hongkang Yang**[4], **Yaoyu Zhang**[1,2, *], **Zhi-Qin John Xu**[1,2,3,*]

[1] Institute of Natural Sciences, School of Mathematical Sciences, Shanghai Jiao Tong University
[2] MOE-LSC, School of Artificial Intelligence, Shanghai Jiao Tong University
[3] Center for LLM, Institute for Advanced Algorithms Research, Shanghai
[4] MemTensor (Shanghai) Technology Co., Ltd.
[†] Equal contribution, list in alphabetical order.

Abstract

Loss spikes emerge commonly during training across neural networks of varying architectures and scales when using the Adam optimizer. In this work, we investigate the underlying mechanism responsible for Adam spikes. While previous explanations attribute these phenomena to the lower-loss-as-sharper characteristics of the loss landscape, our analysis reveals that Adam's adaptive preconditioners themselves can trigger spikes. Specifically, we identify a critical regime where squared gradients become substantially smaller than the second-order moment estimates, causing the latter to undergo a β_2-exponential decay and to respond sluggishly to current gradient information. This mechanism can push the maximum eigenvalue of the preconditioned Hessian beyond the classical stability threshold $2/\eta$ for a sustained period, inducing instability. This instability further leads to an alignment between the gradient and the maximum eigendirection, and a loss spike occurs precisely when the gradient-directional curvature exceeds $2/\eta$. We verify this mechanism through extensive experiments on fully connected networks, convolutional networks, and Transformer architectures.

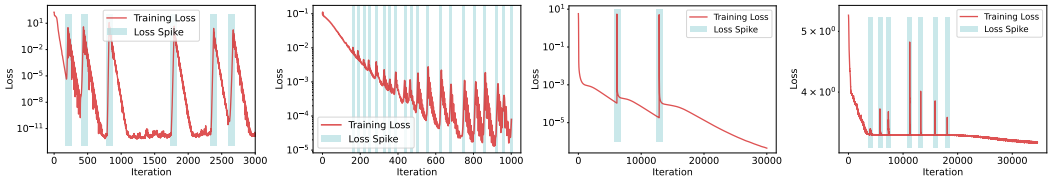
1 Introduction

Neural network optimization remains a complex and sometimes unpredictable process despite significant advances in training methodologies. One particularly intriguing phenomenon that practitioners frequently encounter but rarely explore systematically is the "loss spike" — a sudden and sharp surge in the loss function that subsequently subsides, as illustrated in Fig. 1. These spikes are observed across a wide range of network architectures and datasets, yet their underlying mechanisms remain elusive. Practitioners face a critical dilemma when encountering loss spikes: should they intervene by adjusting hyperparameters to eliminate these apparent anomalies, or might these spikes actually serve some beneficial purpose in the optimization process? Answering these questions requires a deeper theoretical understanding of when, how and why loss spikes occur.

Previous research has tried to explain loss spikes through the geometry of loss landscapes (Ma et al., 2022; Li et al., 2025). The lower-loss-as-sharper (LLAS) hypothesis (Li et al., 2025) suggests that regions of lower loss correspond to sharper curvature in the loss landscape, potentially causing instability. While this explanation provides some intuition, it fails to explain the specific behavior of adaptive optimizers like Adam (Kingma and Ba, 2014) that consistently exhibit spikes even in simple scenarios where landscape geometry is well-understood. For instance, as shown in Fig. 2(a), Adam produces loss spikes on a simple quadratic function even with learning rates well below theoretical stability thresholds, while gradient descent converges smoothly. This behavior can not be explained

*Corresponding author: xuzhiqin@sjtu.edu.cn, zhyy.sjtu@sjtu.edu.cn

Preprint. Under review.



(a) FNN (b) CNN (c) Transformer (d) Transformer

Figure 1: Loss spikes across architectures: (a) FNNs for function approximation. (b) CNNs on CIFAR10. (c-d) Transformers on sequence learning. See experimental details in Appendix E.

by loss landscape alone, since quadratic functions have constant curvature. Furthermore, although prior research has established that training instabilities can occur when the maximum eigenvalue of Hessian or preconditioned Hessian exceeds $2/\eta$ (η is the learning rate) (Cohen et al., 2021; Wu et al., 2018; Xing et al., 2018; Ahn et al., 2022; Lyu et al., 2022; Arora et al., 2022; Wang et al., 2022; Cohen et al., 2023), the precise relationship between such instabilities and observed loss spikes remains unclear. In particular, instability may sometimes manifest as oscillations and sometimes as spikes (Ma et al., 2022), the specific mechanism under which spikes occur is not well understood.

In this paper, we present a detailed mechanistic explanation for loss spikes in Adam optimization. Our key insight is that these spikes arise not primarily from the complex geometry of the loss landscape, but rather from the intrinsic dynamics of Adam's adaptive preconditioners. Specifically, we identify a critical regime where diminishing gradients become substantially smaller than the corresponding second-moment estimates. When this occurs, the second-moment estimates begin an exponential decay governed by β_2, rather than responding to the current gradient information. This decoupling pushes the maximum eigenvalue of the preconditioned Hessian beyond the threshold $2/\eta$ for a sustained period. This instability further leads to an alignment between gradient and maximum eigendirection, and a loss spike occurs precisely when the gradient-directional curvature exceeds $2/\eta$.

Our main contributions are summarized as follows:

(i) We show that Adam's adaptive preconditioners can independently induce training instability by causing the maximum eigenvalue of the preconditioned Hessian $\hat{\boldsymbol{H}}_t$ to exceed the stability threshold. This mechanism is distinct from the lower-loss-as-sharper (LLAS) landscape hypothesis (Li et al., 2025) (please refer to Sec. 3 and Sec. 4.1).

(ii) We identify a critical regime where gradients become significantly smaller than their second-moment estimates when employing a relatively large β_2. This renders the preconditioners insensitive to current gradient information and causes the maximum eigenvalue of the preconditioned Hessian to **persistently** exceed the classical stability bound $2/\eta$ (please refer to Sec. 4.2 and Sec. 5).

(iii) We propose a novel predictor for loss spikes based on the gradient-directional curvature, denoted λ_{grad}, and empirically demonstrate that the condition $\lambda_{\max}(\hat{\boldsymbol{H}}_t) > 2/\eta$ alone is insufficient; a spike occurs specifically when the curvature in the gradient direction exceeds this threshold (please refer to Sec. 4.3 and Sec. 5).

2 Related Work

Edge of Stability (EoS). Various works (Cohen et al., 2021; Wu et al., 2018; Xing et al., 2018; Ahn et al., 2022; Lyu et al., 2022; Arora et al., 2022; Jastrzebski et al., 2020; Jastrzębski et al., 2019; Lewkowycz et al., 2020) have investigated the *Edge of Stability* (EoS), a phenomenon where gradient descent progressively increases the sharpness of the loss landscape—a process known as *progressive sharpening*—until the maximum Hessian eigenvalue stabilizes near the threshold $2/\eta$, while the loss continues to decrease non-monotonically. Ma et al. (2022) proposed a subquadratic structure near local minima, where sharpness increases when the loss decreases along the gradient direction, providing a theoretical account of this behavior. Other studies (Damian et al., 2023; Wang et al., 2022) show that when $\lambda_{\max} > 2/\eta$, self-stabilization mechanisms can reduce sharpness and restore stability. More recently, Cohen et al. (2023) extended the EoS framework to adaptive optimizers,

introducing the concept of *Adaptive Edge of Stability* (AEoS). While EoS has been widely explored, its direct association with loss spikes has yet to be thoroughly investigated.

Convergence Analysis of Adam. Numerous works have analyzed the convergence behavior of adaptive gradient methods (Chen et al., 2019; Li and Orabona, 2019; Xie et al., 2020; Défossez et al., 2022; Da Silva and Gazeau, 2020; Shi et al., 2021; Zou et al., 2019; Zhou et al., 2024). In particular, Reddi et al. (2018) demonstrated that Adam may fail to converge even in simple convex settings, prompting a series of variants (Liu et al., 2019; Taniguchi et al., 2024). Zhang et al. (2022) showed that Adam can converge to a neighborhood of critical points when β_2 is large, and this convergence is guaranteed if $\beta_1 < \sqrt{\beta_2}$.

Loss Spike Analysis. Chowdhery et al. (2023) reported that restarting training from an earlier checkpoint and skipping the spiking data batch can mitigate spikes in large models. Molybog et al. (2023) found that the gradient and second-moment estimates of shallow layer parameters can decay to near-zero and then spike upon encountering a large gradient. Li et al. (2025) argued that spikes occur in sharp regions of the loss landscape with a lower-loss-as-sharper (LLAS) structure. Ma et al. (2022) qualitatively demonstrated that Adam's hyperparameters impact the occurrence of spikes or oscillations. More recently, Cattaneo and Shigida (2025) empirically found that reducing β_2 can effectively mitigate loss spikes. Although previous studies have uncovered parts of the puzzle surrounding spikes, this work provides a more detailed understanding of the spike formation.

3 Distinct Loss Spike Mechanism in Adam vs. Gradient Descent (GD)

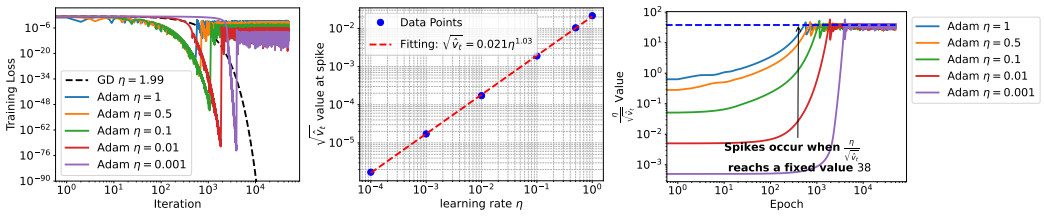
Adam Algorithm. The Adam algorithm is widely used in training Transformer models and is usually more prone to cause loss spikes. Adam maintains exponential moving averages of gradients (first moment) and squared gradients (second moment) to speed up training:

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t, \quad \boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2. \tag{1}$$

where $\boldsymbol{g}_t := \nabla L(\boldsymbol{\theta}_t)$ is the gradient, and $\beta_1, \beta_2 \in [0, 1)$ are hyperparameters controlling the exponential decay rates (default values: $\beta_1 = 0.9, \beta_2 = 0.999$). To counteract the initialization bias toward zero, these moments are corrected: $\hat{\boldsymbol{m}}_t = \frac{\boldsymbol{m}_t}{1-\beta_1^t}, \quad \hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1-\beta_2^t}$. The parameter update rule for Adam is:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\hat{\boldsymbol{m}}_t}{\sqrt{\hat{\boldsymbol{v}}_t} + \varepsilon}. \tag{2}$$

where $\eta > 0$ is the learning rate and $\varepsilon > 0$ is a small constant (default 10^{-8} in PyTorch).



(a) Loss (b) $\sqrt{\hat{v}_t} \propto \eta$ (c) Evolution of $\frac{\eta}{\sqrt{\hat{v}_t}}$

Figure 2: Optimization of $f(\theta) = \frac{1}{2}\theta^2$. (a) Loss trajectories during Adam and GD training across various learning rates. Curves of different colors represent Adam's training loss, which initially decreases steadily before abruptly spiking to significantly higher values. (b) The relationship between learning rate and $\sqrt{\hat{v}_t}$ value at spike occurrence follows a power law, appearing as a straight line with a slope of approximately 1 in log-log scale. (c) Under different learning rates, the ratio $\eta/\sqrt{\hat{v}_t}$ consistently reaches a nearly identical threshold value immediately before the loss begins to spike.

Differences in Spike Behavior Between GD and Adam. Adaptive gradient methods like Adam exhibit fundamentally different behavior compared to standard gradient descent. A notable distinction is that Adam can encounter convergence difficulties even with simple quadratic functions and very small learning rates. For the quadratic function $f(\theta) = \frac{1}{2}\theta^2$, it is well established that gradient descent converges when the learning rate $\eta < 2/\lambda_{\max} = 2$ (depicted by the black dashed line in

Fig. 2(a)). However, Adam displays more intricate dynamics. As illustrated in Fig. 2(a), Adam with a learning rate $\eta \ll 2$ (using hyperparameters $\beta_1 = 0.9, \beta_2 = 0.99, \varepsilon = 10^{-8}$) still fails to converge. This non-convergence manifests in the distinctive colored curves in Fig. 2(a), where the training loss initially decreases steadily before abruptly spiking to a substantially higher magnitude. Fig. 2(b) further examines the relationship between Adam's second moment $\sqrt{\hat{v}_t}$ at spike occurrence and learning rate. From Fig. 2(b), we observe that smaller learning rates correspond to smaller $\sqrt{\hat{v}_t}$ values when spikes occur, with the relationship appearing linear in log-log scale with a slope near 1. For one-dimensional quadratic optimization, $\eta/\sqrt{\hat{v}_t}$ can be interpreted as the actual effective learning rate and it increases as training progresses because $\sqrt{\hat{v}_t}$ diminishes alongside the gradient g_t according to Eq. (1). Experimentally, Fig. 2(c) confirms that this ratio increases until reaching a nearly consistent threshold value 38 (see Lem. 1 for a theoretical explanation), at which point the loss spike invariably occurs. While straightforward, this analysis provides valuable intuition for the emergence of spikes. However, it is important to note that in high-dimensional optimization scenarios, $\sqrt{\hat{\boldsymbol{v}}_t}$ becomes a vector rather than a scalar, rendering the notion of an equivalent learning rate inapplicable. In the following section, we will quantitatively characterize Adam's spike behavior in more general settings.

4 Loss Spike Analysis Based on Quadratic Approximation

Quadratic Approximation. To understand the mechanics behind loss spikes, we first establish a theoretical analysis that connects optimization dynamics with the geometry of the loss landscape. Consider a neural network optimization problem where we aim to minimize a loss function $L(\boldsymbol{\theta})$ with respect to parameters $\boldsymbol{\theta} \in \mathbb{R}^M$. Around any point $\boldsymbol{\theta}$ in parameter space, we can approximate the loss function using a second-order Taylor expansion with Lagrangian remainder $L(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \nabla L(\boldsymbol{\theta})^\top \delta\boldsymbol{\theta} + \frac{1}{2}\delta\boldsymbol{\theta}^\top \boldsymbol{H}(\boldsymbol{\theta}')\delta\boldsymbol{\theta}$, where $\nabla L(\boldsymbol{\theta}) \in \mathbb{R}^M$ is the gradient vector and $\boldsymbol{H}(\boldsymbol{\theta}') = \nabla^2 L(\boldsymbol{\theta}') \in \mathbb{R}^{M\times M}$ is the Hessian matrix of second derivatives evaluated at $\boldsymbol{\theta}'$, with $\boldsymbol{\theta}' \in (\boldsymbol{\theta}, \boldsymbol{\theta} + \delta\boldsymbol{\theta})$. The Hessian characterizes the local curvature of the loss landscape. Although deep neural network loss functions are highly non-convex with respect to parameters $\boldsymbol{\theta}$ and therefore not globally quadratic, when $\delta\boldsymbol{\theta}$ is sufficiently small and the loss function is smooth, the Hessian $\boldsymbol{H}$ remains approximately constant in the local region. Under these conditions, the second-order approximation simplifies to:

$$L(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) \approx \tilde{L}(\delta\boldsymbol{\theta}) := L(\boldsymbol{\theta}) + \nabla L(\boldsymbol{\theta})^\top \delta\boldsymbol{\theta} + (1/2)\delta\boldsymbol{\theta}^\top \boldsymbol{H}\delta\boldsymbol{\theta}. \tag{3}$$

Stability Analysis Based on Quadratic Approximation. In standard gradient descent with learning rate η, the parameter update follows: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\nabla L(\boldsymbol{\theta}_t)$. Assume the second-order Taylor expansion in Eq. (3) is valid, then for a small perturbation $\delta\boldsymbol{\theta}_t$ around $\boldsymbol{\theta}$, we have:

$$\delta\boldsymbol{\theta}_{t+1} \approx \delta\boldsymbol{\theta}_t - \eta\nabla\tilde{L}(\delta\boldsymbol{\theta}_t) = \delta\boldsymbol{\theta}_t - \eta(\nabla L(\boldsymbol{\theta}) + \boldsymbol{H}\delta\boldsymbol{\theta}_t) = (\boldsymbol{I} - \eta\boldsymbol{H})\delta\boldsymbol{\theta}_t - \eta\nabla L(\boldsymbol{\theta}). \tag{4}$$

When $\lambda_{\max}(\boldsymbol{H}) > 2/\eta$, the iteration becomes unstable along the maximum eigendirection.

4.1 Modified Stability Analysis for Adam

Stability Analysis of Adaptive Mechanism. To analyze the stability conditions of Adam, we first examine solely the adaptive mechanism by setting $\beta_1 = 0$, thus ignoring momentum effects. Following an approach similar to standard gradient descent analysis, if the second-order Taylor expansion in Eq. (3) holds, then for a small perturbation $\delta\boldsymbol{\theta}$ around $\boldsymbol{\theta}$, we have:

$$\delta\boldsymbol{\theta}_{t+1} \approx \delta\boldsymbol{\theta}_t - \eta\frac{\nabla\tilde{L}(\delta\boldsymbol{\theta}_t)}{\sqrt{\hat{\boldsymbol{v}}_t} + \varepsilon} = \left(\boldsymbol{I} - \eta\text{diag}\left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_t} + \varepsilon}\right)\boldsymbol{H}\right)\delta\boldsymbol{\theta}_t - \eta\frac{\nabla L(\boldsymbol{\theta})}{\sqrt{\hat{\boldsymbol{v}}_t} + \varepsilon}. \tag{5}$$

Analogous to Eq. (4), stability of this iteration requires the spectral radius $\rho\left(\boldsymbol{I} - \eta\hat{\boldsymbol{H}}\right)$ to be less than 1, where $\hat{\boldsymbol{H}} = \text{diag}\left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_t}+\varepsilon}\right)\boldsymbol{H}$ is the "adaptive preconditioned Hessian" of Adam, consistent with previous literature (Cohen et al., 2023). This directly yields the stability condition $\rho(\hat{\boldsymbol{H}}) < 2/\eta$. Although $\hat{\boldsymbol{H}} = \text{diag}\left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_t}+\varepsilon}\right)\boldsymbol{H}$ is asymmetric, it can still be diagonalized and possesses real eigenvalues (see Appendix B Lem. B.1). Therefore, the stability condition becomes $\lambda_{\max}(\hat{\boldsymbol{H}}) < 2/\eta$.

Stability Analysis of Momentum Mechanism. When momentum is introduced ($\beta_1 > 0$), we can analyze the momentum mechanism independently from the adaptive mechanism, considering the

update rule $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{m}_t$ where $\boldsymbol{m}_t$ is first-order momentum. Following the second-order Taylor expansion approach, we have:

$$\delta\boldsymbol{\theta}_{t+1} \approx \delta\boldsymbol{\theta}_t - \eta(\beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\nabla\tilde{L}(\delta\boldsymbol{\theta}_t)) = \delta\boldsymbol{\theta}_t - \eta(\beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)(\nabla L(\boldsymbol{\theta}) + \boldsymbol{H}\delta\boldsymbol{\theta}_t)).$$

Substituting $\eta\boldsymbol{m}_{t-1} = \delta\boldsymbol{\theta}_{t-1} - \delta\boldsymbol{\theta}_t$, we obtain:

$$\delta\boldsymbol{\theta}_{t+1} \approx \left[(1+\beta_1)\boldsymbol{I} - \eta(1-\beta_1)\boldsymbol{H}\right]\delta\boldsymbol{\theta}_t - \beta_1\delta\boldsymbol{\theta}_{t-1} - \eta(1-\beta_1)\nabla L(\boldsymbol{\theta}). \tag{6}$$

The stability condition for this three-term recursion is given in Lem. 1.

Lemma 1 (see Appendix B Lem. B.2 for proof)**.** *The three-term recursive iteration* (6) *converges if and only if* $\lambda_{\max}(\frac{1-\beta_1}{1+\beta_1}\boldsymbol{H}) < 2/\eta$.

Comprehensive Stability Analysis of Adam. When considering the complete update formula of Adam, Eq. (2), both the adaptive mechanism and the momentum mechanism should be integrated. Additionally, when incorporating the momentum bias correction term $\hat{\boldsymbol{m}}_t = \frac{\boldsymbol{m}_t}{1-\beta_1^t}$, the comprehensive "Adam preconditioned Hessian" becomes:

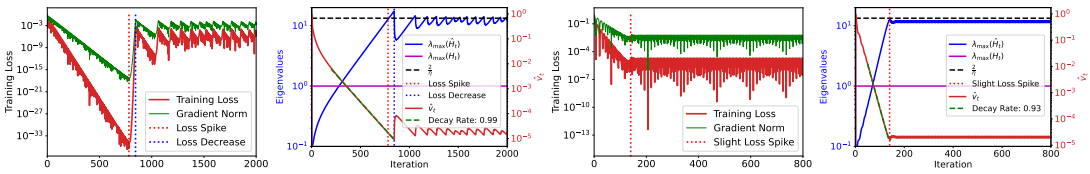
$$\hat{\boldsymbol{H}}_t = \frac{1}{1-\beta_1^t}\frac{1-\beta_1}{1+\beta_1}\mathrm{diag}\left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_t}+\varepsilon}\right)\boldsymbol{H}_t. \tag{7}$$

In the subsequent sections, we experimentally validate that this modified stability criterion $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ accurately corresponds to the occurrence of loss spikes in practical optimization scenarios.

4.2 Adaptive Preconditioners Trigger Loss Spike

The key difference of the stability condition between gradient descent and Adam is the adaptive preconditioners $\boldsymbol{v}_t$. To investigate the effect of the decay behavior of $\boldsymbol{v}_t$ on loss spikes, we conducted controlled experiments on a simple quadratic objective $f(\theta) = \frac{1}{2}\theta^2$. Fig. 3(a–b) shows results under the Adam setting with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. Initially, the loss decreases smoothly. However, a loss spike occurs at epoch 782, precisely when the maximum eigenvalue of the preconditioned Hessian, $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$, exceeds the critical threshold $2/\eta$.

Fig. 3(a) shows the evolution of the gradient norm (green line), while Fig. 3(b) plots the second-order moment estimate $\hat{v}_t$ (red line). Notably, the gradient norm ($\approx 10^{-15}$) becomes very small before the spike—much smaller than $\sqrt{\hat{v}_t}$ ($\approx 10^{-1}$). According to the update rule (Eq. (1)), this leads the training to enter a regime where v_t decays exponentially as $v_t \approx \beta_2 v_{t-1}$. The green dashed line in Fig. 3(b) fits this decay using $\hat{v}_t = A\alpha^t$, showing excellent agreement with the actual $\hat{v}_t$, and confirming $\alpha \approx \beta_2 = 0.99$. When $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ surpasses $2/\eta$, a loss spike occurs and the gradient norm g_t begins to increase. However, the condition $g_t \ll \sqrt{\hat{v}_t}$ persists, causing the exponential decay of v_t to continue. This sustained decay consequently maintains the elevation of $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ above the stability threshold $2/\eta$ over time. As the spike progresses, the gradient norm eventually grows large enough to impact v_t, at which point $\hat{v}_t$ begins to increase rapidly. This causes $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ to drop back below $2/\eta$, and the loss begins to decrease again at epoch 845.



(a) $\beta_2 = 0.99$ (b) $\beta_2 = 0.99$ (c) $\beta_2 = 0.9$ (d) $\beta_2 = 0.9$

Figure 3: Adam optimization on $f(\theta) = \frac{1}{2}\theta^2$ with different β_2 values. (a, c) Evolution of training loss and gradient norm. (b, d) Evolution of the second moment estimate $\hat{\boldsymbol{v}}_t$ and the maximum eigenvalue of the preconditioned Hessian. The red dotted line marks the onset of the loss spike, while the blue dotted line indicates the point where the loss begins to decrease. The green dashed lines fit $\hat{\boldsymbol{v}}_t$ decay using $\hat{\boldsymbol{v}}_t = A\alpha^t$ with decay rate shown in the labels.

In contrast, employing a smaller β_2 increases v_t's sensitivity to gradient changes and may alter this behavior. Fig. 3(c–d) present results for $\beta_1 = 0.9$ and $\beta_2 = 0.9$—a configuration less commonly used

in practice due to its inferior convergence guarantees (Shi et al., 2021; Zhang et al., 2022). In this setting, the gradient remains non-negligible relative to $\sqrt{v_t}$ throughout training, effectively preventing the onset of β_2-exponential decay (e.g., the observed decay rate $\alpha \approx 0.93$ in Fig. 3(d) is larger than $\beta_2 = 0.9$). As training progresses, the gradient gradually diminishes and $\hat{v}_t$ continues to decrease, which leads to a gradual increase in $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$. However, since the gradient is non-negligible, once $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ reaches the critical threshold $2/\eta$, the gradient norm begins to rise, causing an immediate adjustment in $\boldsymbol{v}_t$. This feedback mechanism prevents $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ from persistently exceeding the stability threshold, thereby suppressing the emergence of pronounced loss spikes. As illustrated in Fig. 3(c), the loss exhibits a minor rise followed by oscillations, never reaching a large spike. This helps explain why Adam training, as empirically observed by Ma et al. (2022), sometimes results in sudden spikes in loss and sometimes in oscillatory behavior.

4.3 Precise Loss Spike Prediction via Gradient-Directional Curvature

In high-dimensional optimization, when the maximum eigenvalue of the Hessian satisfies $\lambda_{\max} > 2/\eta$, instability arises primarily along the corresponding eigendirection, while the remaining directions may still exhibit stable descent. As a result, a loss spike does not necessarily occur immediately, with not even any visible signs of abnormality (see Fig. 4(a)). To more precisely predict the onset of a loss spike, we analyze the change in the loss value between consecutive optimization steps. Applying a second-order Taylor expansion of the loss function L at $\boldsymbol{\theta}_t$, we obtain: $L(\boldsymbol{\theta}_{t+1}) \approx L(\boldsymbol{\theta}_t) + \nabla L(\boldsymbol{\theta}_t)^\top(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t) + \frac{1}{2}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)^\top \boldsymbol{H}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)$. Substituting the gradient descent update rule $\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t = -\eta \nabla L(\boldsymbol{\theta}_t)$, the estimated loss change becomes: $L(\boldsymbol{\theta}_{t+1}) - L(\boldsymbol{\theta}_t) \approx -\eta \|\nabla L(\boldsymbol{\theta}_t)\|^2 + \frac{1}{2}\eta^2 \nabla L(\boldsymbol{\theta}_t)^\top \boldsymbol{H} \nabla L(\boldsymbol{\theta}_t)$. Assuming the quadratic approximation holds, an increase in loss—i.e., a necessary condition for a spike to occur when:

$$\lambda_{\text{grad}}(\boldsymbol{H}) := \frac{\nabla L(\boldsymbol{\theta}_t)^\top \boldsymbol{H} \nabla L(\boldsymbol{\theta}_t)}{\|\nabla L(\boldsymbol{\theta}_t)\|^2} > \frac{2}{\eta}. \tag{8}$$

Here, λ_{grad} denotes the curvature of the loss landscape along the gradient direction. A loss spike is therefore predicted only when the gradient becomes sufficiently aligned with the dominant curvature direction. For Adam, where the Hessian is preconditioned, we analogously define the predictor as $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}) := \frac{\nabla L(\boldsymbol{\theta}_t)^\top \hat{\boldsymbol{H}} \nabla L(\boldsymbol{\theta}_t)}{\|\nabla L(\boldsymbol{\theta}_t)\|^2}$, where $\hat{\boldsymbol{H}}$ denotes the preconditioned Hessian in Eq. (7).

Experimental Verification of Loss Spike Predictor. We validate the proposed loss spike predictor using a two-layer fully connected neural network trained on 20 data points to fit the 1-dimensional target function $f(x) = \sin(x) + \sin(4x)$ (see Appendix E for experimental details). The model is trained using either gradient descent or Adam with full-batch. During training, we track both $\lambda_{\max}(\boldsymbol{H}_t)$ and $\lambda_{\text{grad}}(\boldsymbol{H}_t)$. For gradient descent, as shown in Fig. 4(a–b), two prominent loss spikes are observed. At epoch 416, although $\lambda_{\max}(\boldsymbol{H}_t)$ already exceeds $2/\eta$, the loss continues to decrease. A sharp loss increase (spike) at epoch 580 occurs only when $\lambda_{\text{grad}}(\boldsymbol{H}_t)$ also exceeds $2/\eta$. Once $\lambda_{\text{grad}}(\boldsymbol{H}_t)$ falls below the threshold, the loss resumes decreasing. Notably, during the initial two epochs, $\lambda_{\max}(\boldsymbol{H}_t)$ and $\lambda_{\text{grad}}(\boldsymbol{H}_t)$ also exceed $2/\eta$ transitorily without triggering any spikes. This period corresponds to rapid loss decrease, suggesting that the Hessian varies rapidly and the quadratic approximation assumption may not hold during this phase. For Adam, Fig. 4(c–d) shows 7 distinct loss spikes. However, $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ exceeds $2/\eta$ at 10 different time steps. Crucially, spikes occur only when $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t) > 2/\eta$, confirming that $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ alone is insufficient to predict spikes.

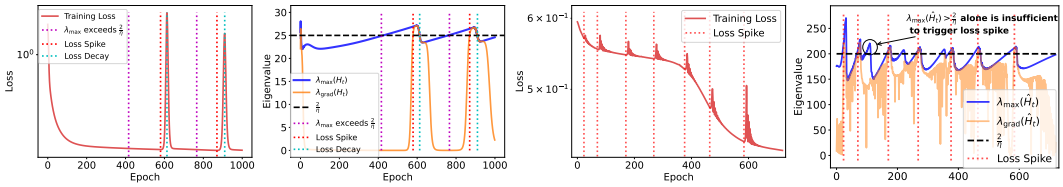
4.4 The Mechanics of Loss Spike Formation in Adam

Building on our theoretical and empirical findings, we identify a five-phase progression that characterizes the formation and resolution of loss spikes during training with the Adam optimizer.

Phase 1: Stable Loss Decrease. Training loss decreases steadily with no abnormalities observed.

Phase 2: Decay of the Adaptive Preconditioners. As the gradient $\boldsymbol{g}_t$ diminishes for some layers, the corresponding second-moment estimate $\boldsymbol{v}_t$ begins to decay. Under typical settings with large $\beta_2 \in [0.95, 0.9999]$, $\|\boldsymbol{g}_t\|$ can be much smaller than $\|\sqrt{\boldsymbol{v}_t}\|$, causing $\boldsymbol{v}_t$ to enter an β_2-dominant exponential decay regime: $\boldsymbol{v}_t \approx \beta_2 \boldsymbol{v}_{t-1}$. This decay reduces the strength of the adaptive preconditioners $\boldsymbol{v}_t$.

Phase 3: Onset of the Loss Spike. Instability arises when the maximum eigenvalue of the preconditioned Hessian, $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$, exceeds the stability threshold $2/\eta$. Initially localized, the instability



(a) GD (loss) (b) GD (eigenvalues) (c) Adam (loss) (d) Adam (eigenvalues)

Figure 4: Experimental validation of the gradient-directional loss spike predictor. A two-layer fully connected neural network (width $1,000$, approximately $3,000$ parameters) is trained on 200 randomly sampled data points to fit $f(x) = \sin(x) + \sin(4x)$. (a–b) Gradient descent with learning rate $\eta = 0.08$. (c–d) Adam with learning rate $\eta = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$.

intensifies as the gradient aligns with the unstable curvature direction. A loss spike occurs only when the gradient-projected curvature λ_{grad} also surpasses $2/\eta$. Since $\boldsymbol{v}_t$ responds sluggishly to current gradient information $\boldsymbol{g}_t$, λ_{grad} will persistently exceed $2/\eta$.

Phase 4: Growth of the Adaptive Preconditioners. As the loss spike intensifies, the gradient norm grows progressively larger. When the gradient becomes sufficiently large to influence $\sqrt{\boldsymbol{v}_t}$, the decay of $\boldsymbol{v}_t$ halts and reverses. The resulting growth in $\boldsymbol{v}_t$ reduces $\lambda_{\text{grad}}(\hat{\boldsymbol{H}})$, helping to restore stability.

Phase 5: Loss Decay Phase: When $\lambda_{\text{grad}}(\hat{\boldsymbol{H}})$ falls back below $2/\eta$, the optimizer regains stability. The loss resumes decreasing, completing the spike cycle and returning to Phase 1.

These five phases provide a comprehensive intuitive understanding of the Adam loss spike phenomenon. Furthermore, we also provide a mathematically rigorous characterization of these phases for a one-dimensional quadratic optimization in Appendix B Thm. B.1.

5 Loss Spike Analysis in Neural Network Optimization

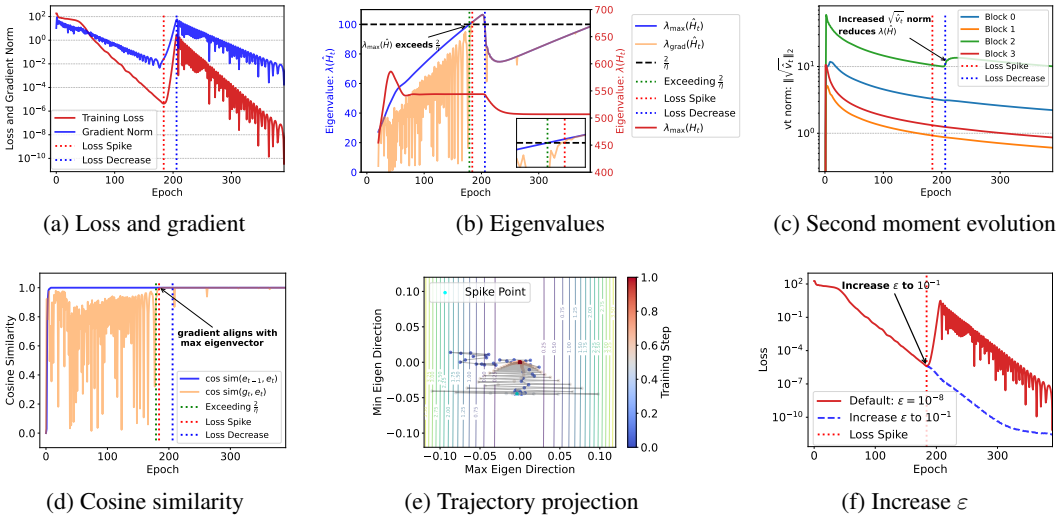
To validate our proposed spike mechanism and evaluate our predictors' effectiveness in high-dimensional, non-convex settings, we performed empirical studies across various neural network architectures and tasks. Detailed experimental configurations are provided in Appendix E, with supplementary experiments presented in Appendix D.

5.1 Fully Connected Neural Networks for Function Approximation

We trained a two-layer fully connected network on a 50-dimensional function approximation task using Adam hyperparameters $\beta_1 = 0.9, \beta_2 = 0.999$. Fig. 5(a) shows optimization dynamics mirroring our quadratic function analysis: both loss and gradient norm decrease rapidly before experiencing a sharp spike. We track maximum eigenvalue evolution of Hessian and the preconditioned Hessian during training. Fig. 5(b) shows $\lambda_{\max}(\boldsymbol{H}_t)$ quickly stabilizing while $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ continues to increases due to the decrease of $\boldsymbol{v}_t$ in Fig. 5(c). Though $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ surpasses the stability threshold $2/\eta$ at epoch 179, the spike occurs at epoch 184, precisely when $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ exceeds $2/\eta$ (Fig. 5(b)).

Fig. 5(c) illustrates the evolution of second-moment norms $\sqrt{\hat{\boldsymbol{v}}_t}$ for each parameter block. Before the spike, gradient norm $\|\boldsymbol{g}_t\|$ ($\approx 10^{-2}$) becomes significantly smaller than $\|\sqrt{\hat{\boldsymbol{v}}_t}\|$, causing $\boldsymbol{v}_t$ to decay exponentially at rate β_2. After spike onset, the gradient norm increases, while $\hat{\boldsymbol{v}}_t$ continues to decrease due to its sluggish response. Once the gradient norm becomes sufficiently large, $\boldsymbol{v}_t$ begins to rise rapidly, which drives $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ below $2/\eta$, allowing the loss to resume its descent at epoch 206.

The cosine similarity between maximum eigenvectors of $\boldsymbol{H}_t$ across consecutive steps approaches 1 early in training (Fig. 5(d)), validating our quadratic approximation and loss spikes occur when gradient aligns with maximum curvature direction. Fig. 5(e) confirms this by projecting the trajectory onto maximum and minimum eigenvectors. Intuitively, pre-spike optimization resembles traversing a river valley; when $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ violates stability, oscillations along the valley direction generate the



(a) Loss and gradient (b) Eigenvalues (c) Second moment evolution

(d) Cosine similarity (e) Trajectory projection (f) Increase ε

Figure 5: (a) Training loss and gradient norm over time. (b) Evolution of critical eigenvalues: original Hessian maximum eigenvalue $\lambda_{\max}(\boldsymbol{H}_t)$, preconditioned Hessian maximum eigenvalue $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ and gradient-directional eigenvalue $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ relative to $2/\eta$. (c) L_2-norm of second moment $||\sqrt{\hat{\boldsymbol{v}}_t}||_2$ of different parameter blocks during training. (d) Cosine similarity between maximum eigenvectors in two consecutive epochs (blue) and between gradient and current maximum eigenvector (orange). (e) Training trajectory projected onto maximum and minimum Hessian eigenvectors at epoch 390. The colorbar for training steps is normalized to the range $[0, 1]$, where 0 corresponds to epoch 28 and 1 corresponds to epoch 390, to better visualize the trajectory near the spike. (f) Increase the default ε in Eq. (2) to 0.1 at epoch 184.

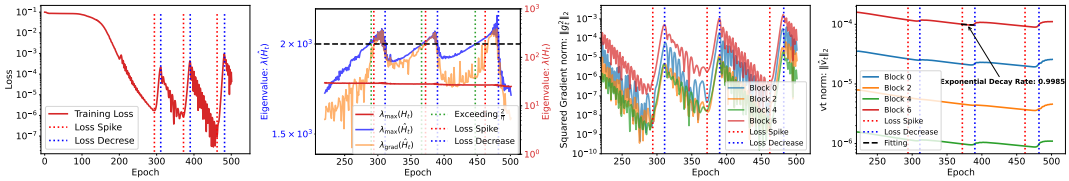
spike. To suppress the spike, a straightforward method involves increasing ε in Eq. (2). As shown in Fig. 5(f), increasing ε to 0.1 at spike onset effectively eliminates it.

5.2 Convolutional Neural Networks for Image Classification

We trained a convolutional neural network on CIFAR10 using Adam hyperparameters $\beta_1 = 0.9, \beta_2 = 0.999$. As shown in Fig. 6(a), the optimization follows a pattern similar to FNN, with an initial loss decrease followed by three distinct spikes. Analysis of the preconditioned Hessian's eigenvalues (Fig. 6(b)) shows $\lambda_{\max}(\boldsymbol{H}_t)$ remaining below the stability threshold $2/\eta$, while $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ increases until exceeding it. Loss spikes occur precisely when $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ surpasses $2/\eta$. Figs. 6(c-d) show the evolution of squared gradients and second-order moments $\sqrt{\hat{\boldsymbol{v}}_t}$ across parameter blocks. Before spikes, $\|\boldsymbol{g}_t\|$ is much smaller than $\|\sqrt{\hat{\boldsymbol{v}}_t}\|$, with $\hat{\boldsymbol{v}}_t$ decaying exponentially at rate $\approx \beta_2$. During spikes, while $\hat{\boldsymbol{v}}_t$ continues decreasing, the gradient norm increases until substantially impacting $\boldsymbol{v}_t$. Subsequently, $\hat{\boldsymbol{v}}_t$ rises, causing $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ to fall below $2/\eta$ and allowing loss descent to resume.

5.3 Transformer Models for Sequence Learning

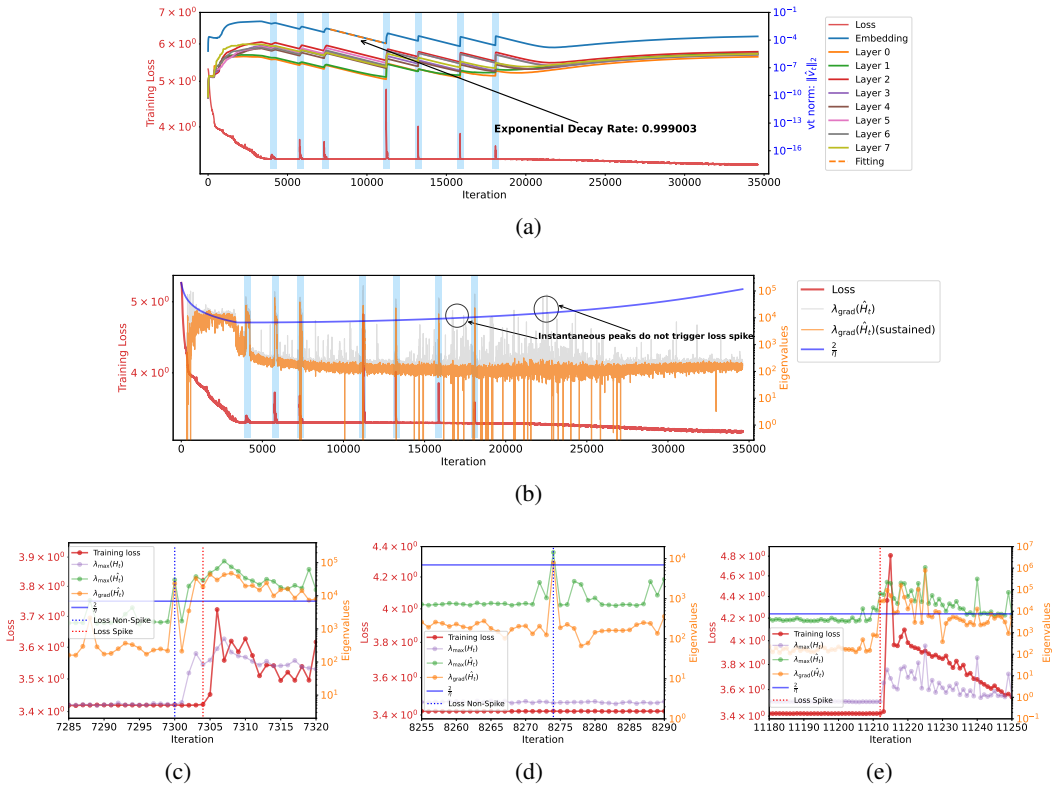
We trained an 8-layer Transformer (approximately 10 million parameters) on a synthetic dataset of 900k sequences (batch size 2048) for compositional rule learning under the next-token prediction paradigm. Fig. 7(a) shows seven distinct loss spikes (blue regions). Prior to each spike, the norm of the second-moment estimate $\hat{\boldsymbol{v}}_t$ for the embedding and $\boldsymbol{W}_V$ parameters across attention layers decays at a rate of approximately 0.999003 (close to β_2), followed by a sudden increase in $\|\hat{\boldsymbol{v}}_t\|$ and a sharp drop in loss. To investigate whether these spikes correspond to the onset of instability, we tracked $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ (Fig. 7(b), gray line). While spikes coincide with $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ exceeding $2/\eta$, not all threshold crossings trigger spikes. A detailed analysis of these events revealed that transient periods where $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ exceeds $2/\eta$ do not necessarily cause a spike. Loss spikes only occur when $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ remains above the threshold for a sustained duration (Fig. 7(c-e)). Consequently, we defined a "sus-



(a) Loss (b) Eigenvalues (c) Squared gradient (d) Second moment

Figure 6: Training a CNN on 50 randomly selected CIFAR-10 images to illustrate the detailed spikes (see similar result for larger datasets in Appendix D Fig. D6). (a) Training loss over time. (b) Evolution of eigenvalues: original Hessian maximum eigenvalue $\lambda_{\max}(\boldsymbol{H}_t)$, preconditioned Hessian maximum eigenvalue $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$, and gradient-directional eigenvalue $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ relative to $2/\eta$ (black dashed line). (c) Gradient norm evolution across parameter blocks. (d) L_2-norm of second moment estimate $\|\hat{\boldsymbol{v}}_t\|$ of different parameter blocks.

tained spike predictor" as: $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)(\text{sustained}) = \min(\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_{t-1}), \lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t), \lambda_{\text{grad}}(\hat{\boldsymbol{H}}_{t+1}))$. This refined predictor ((Fig. 7(b), orange line)) demonstrates perfect correspondence with loss spike occurrences. Sustained periods above threshold trigger loss spikes, which is consistent with the findings in Fig. 3.



(a)

(b)

(c) (d) (e)

Figure 7: (a) Evolution of training loss and second moment $\|\hat{\boldsymbol{v}}_t\|$, with seven spikes highlighted. (b) Gradient-directional eigenvalues $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ (gray) and sustained predictor $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)(\text{sustained})$ (orange) vs. $2/\eta$. (c-e) Detailed inspection of threshold-exceeding intervals showing the maximum eigenvalues of the original Hessian $\lambda_{\max}(\boldsymbol{H}_t)$, preconditioned Hessian $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$, and $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$.

6 Conclusion and Discussion

We present a detailed analysis for loss spikes in Adam, revealing that the adaptive preconditioners themselves can trigger these spikes. However, it is possible that both the geometry of the loss landscape and the preconditioners jointly contribute to loss spikes. Disentangling their individual contributions and attributing different spike mechanisms remains an open direction for future work.

Loss spikes represent more than mere optimization phenomena; they may signify transitions between distinct attractor basins in the landscape. Our experiments in Appendix C identify four spike types (**neutral**, **beneficial**, **malignant**, and **catastrophic**) in Transformer training—highlighting the importance of context-specific decisions on whether to suppress or preserve them. Precisely distinguishing between these spike types remains an unresolved challenge.

When severe spikes disrupt training, several mitigation strategies exist. Increasing ε or β_1 can reduce the preprocessed Hessian, while reducing β_2 (Cattaneo and Shigida, 2025) makes the second-moment more responsive to recent gradients, breaking the persistence condition that leads to spikes. Alternative techniques include sandwich normalization (Ding et al., 2021; Yin et al., 2025), σ-Reparam (Zhai et al., 2023), and scaled-decouple distribution (Wang et al., 2025). While some studies (Lyu et al., 2022; Mueller et al., 2023) attribute normalization's effectiveness to sharpness reduction, a deeper understanding of how to leverage or control spikes remains a promising avenue for future research.

Acknowledgments and Disclosure of Funding

This work is supported by the National Key R&D Program of China Grant No. 2022YFA1008200, the National Natural Science Foundation of China Grant No. 92270001, 12371511, 12422119, Shanghai Municipal of Science and Technology Major Project No. 2021SHZDZX0102, the Fundamental Research Funds for the Central Universities (project number YG2024ZD03), and the HPC of School of Mathematical Sciences and the Student Innovation Center, and the Siyuan-1 cluster supported by the Center for High Performance Computing at Shanghai Jiao Tong University, and Key Laboratory of Marine Intelligent Equipment and System (Ministry of Education, P.R. China), and SJTU Kunpeng & Ascend Center of Excellence.

References

C. Ma, D. Kunin, L. Wu, L. Ying, Beyond the quadratic approximation: The multiscale structure of neural network loss landscapes, Journal of Machine Learning 1 (2022) 247–267. URL: http://global-sci.org/intro/article_detail/jml/21028.html. doi:https://doi.org/10.4208/jml.220404.

X. Li, Z.-Q. J. Xu, Z. Zhang, Loss spike in training neural networks, Journal of Computational Mathematics (2025).

D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

J. Cohen, S. Kaur, Y. Li, J. Z. Kolter, A. Talwalkar, Gradient descent on neural networks typically occurs at the edge of stability, in: International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=jh-rTtvkGeM.

L. Wu, C. Ma, W. E, How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective, Advances in Neural Information Processing Systems 31 (2018).

C. Xing, D. Arpit, C. Tsirigotis, Y. Bengio, A walk with sgd, arXiv preprint arXiv:1802.08770 (2018).

K. Ahn, J. Zhang, S. Sra, Understanding the unstable convergence of gradient descent, in: International conference on machine learning, PMLR, 2022, pp. 247–257.

K. Lyu, Z. Li, S. Arora, Understanding the generalization benefit of normalization layers: Sharpness reduction, Advances in Neural Information Processing Systems 35 (2022) 34689–34708.

S. Arora, Z. Li, A. Panigrahi, Understanding gradient descent on the edge of stability in deep learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 948–1024.

Z. Wang, Z. Li, J. Li, Analyzing sharpness along gd trajectory: Progressive sharpening and edge of stability, Advances in Neural Information Processing Systems 35 (2022) 9983–9994.

J. Cohen, B. Ghorbani, S. Krishnan, N. Agarwal, S. Medapati, M. Badura, D. Suo, Z. Nado, G. E. Dahl, J. Gilmer, Adaptive gradient methods at the edge of stability, in: NeurIPS 2023 Workshop Heavy Tails in Machine Learning, 2023.

C. Ma, L. Wu, w. E, A qualitative study of the dynamic behavior for adaptive gradient algorithms, in: Mathematical and scientific machine learning, PMLR, 2022, pp. 671–692.

S. Jastrzebski, M. Szymczak, S. Fort, D. Arpit, J. Tabor, K. Cho*, K. Geras*, The break-even point on optimization trajectories of deep neural networks, in: International Conference on Learning Representations, 2020. URL: https://openreview.net/forum?id=r1g87C4KwB.

S. Jastrzębski, Z. Kenton, N. Ballas, A. Fischer, Y. Bengio, A. Storkey, On the relation between the sharpest directions of DNN loss and the SGD step length, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=SkgEaj05t7.

A. Lewkowycz, Y. Bahri, E. Dyer, J. Sohl-Dickstein, G. Gur-Ari, The large learning rate phase of deep learning: the catapult mechanism, arXiv preprint arXiv:2003.02218 (2020).

A. Damian, E. Nichani, J. D. Lee, Self-stabilization: The implicit bias of gradient descent at the edge of stability, in: The Eleventh International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=nhKHA59gXz.

X. Chen, S. Liu, R. Sun, M. Hong, On the convergence of a class of adam-type algorithms for non-convex optimization, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=H1x-x309tm.

X. Li, F. Orabona, On the convergence of stochastic gradient descent with adaptive stepsizes, in: The 22nd international conference on artificial intelligence and statistics, PMLR, 2019, pp. 983–992.

Y. Xie, X. Wu, R. Ward, Linear convergence of adaptive stochastic gradient descent, in: International conference on artificial intelligence and statistics, PMLR, 2020, pp. 1475–1485.

A. Défossez, L. Bottou, F. Bach, N. Usunier, A simple convergence proof of adam and adagrad, Transactions on Machine Learning Research (2022). URL: https://openreview.net/forum?id=ZPQhzTSWA7.

A. B. Da Silva, M. Gazeau, A general system of differential equations to model first-order adaptive algorithms, Journal of Machine Learning Research 21 (2020) 1–42.

N. Shi, D. Li, M. Hong, R. Sun, RMSprop converges with proper hyper-parameter, in: International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=3UDSdyIcBDA.

F. Zou, L. Shen, Z. Jie, W. Zhang, W. Liu, A sufficient condition for convergences of adam and rmsprop, in: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, 2019, pp. 11127–11135.

D. Zhou, J. Chen, Y. Cao, Z. Yang, Q. Gu, On the convergence of adaptive gradient methods for nonconvex optimization, Transactions on Machine Learning Research (2024). URL: https://openreview.net/forum?id=Gh0cxhbz3c, featured Certification.

S. J. Reddi, S. Kale, S. Kumar, On the convergence of adam and beyond, in: International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=ryQu7f-RZ.

L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, On the variance of the adaptive learning rate and beyond, in: International Conference on Learning Representations, 2019.

S. Taniguchi, K. Harada, G. Minegishi, Y. Oshima, S. C. Jeong, G. Nagahara, T. Iiyama, M. Suzuki, Y. Iwasawa, Y. Matsuo, Adopt: Modified adam can converge with any β_2 with the optimal rate, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.

Y. Zhang, C. Chen, N. Shi, R. Sun, Z.-Q. Luo, Adam can converge without any modification on update rules, Advances in neural information processing systems 35 (2022) 28386–28399.

A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, Journal of Machine Learning Research 24 (2023) 1–113.

I. Molybog, P. Albert, M. Chen, Z. DeVito, D. Esiobu, N. Goyal, P. S. Koura, S. Narang, A. Poulton, R. Silva, et al., A theory on adam instability in large-scale machine learning, arXiv preprint arXiv:2304.09871 (2023).

M. D. Cattaneo, B. Shigida, Tuning adam(w): Default β_2 may be too large (2025).

M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang, et al., Cogview: Mastering text-to-image generation via transformers, Advances in neural information processing systems 34 (2021) 19822–19835.

Y. Yin, W. Huang, K. Song, Y. Tang, X. Wu, W. Guo, P. Guo, Y. Wang, X. Meng, Y. Wang, D. Li, C. Chen, D. Tu, Y. Li, F. Yu, R. Tang, Y. Wang, B. Wang, B. Wang, B. Wang, B. Liu, C. Zhang, D. Tang, F. Mi, H. Jin, J. Wei, J. Qin, J. Li, J. Zhao, L. Deng, L. Li, M. Xu, N. Zhang, N. Zheng, Q. Li, R. Ruan, S. Cheng, T. Guo, W. He, W. Li, W. Liu, W. Liu, X. Dai, Y. Dong, Y. Pan, Y. Li, Y. Wang, Y. Li, Y. Ni, Z. Liu, Z. Zhang, Z. Liu, Pangu ultra: Pushing the limits of dense large language models on ascend npus, 2025. URL: https://arxiv.org/abs/2504.07866. arXiv:2504.07866.

S. Zhai, T. Likhomanenko, E. Littwin, D. Busbridge, J. Ramapuram, Y. Zhang, J. Gu, J. M. Susskind, Stabilizing transformer training by preventing attention entropy collapse, in: International Conference on Machine Learning, PMLR, 2023, pp. 40770–40803.

Y. Wang, Z. Zhuo, Y. Zeng, X. Zhou, J. Yang, X. Li, Scale-distribution decoupling: Enabling stable and effective training of large language models, arXiv preprint arXiv:2502.15499 (2025).

M. Mueller, T. J. Vlaar, D. Rolnick, M. Hein, Normalization layers are all that sharpness-aware minimization needs, in: Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL: https://openreview.net/forum?id=lArwl3y9x6.

S. Elaydi, An Introduction to Difference Equations, Undergraduate Texts in Mathematics, 3rd ed., Springer Science & Business Media, 2005.

Z. Zhang, Z. Wang, J. Yao, Z. Zhou, X. Li, W. E, Z.-Q. J. Xu, Anchor function: a type of benchmark functions for studying language models, in: ICLR 2025 Workshop Bridging the Gap Between Practice and Theory in Deep Learning, 2025. URL: https://arxiv.org/abs/2401.08309.

A Limitation and Future Work

Our detailed analysis of loss spikes in Adam optimization reveals that adaptive preconditioners can themselves trigger these phenomena and we verify this mechanism in certain neural network architectures. However, we acknowledge that in more complex scenarios, both the intrinsic geometry of the loss landscape and the applied preconditioners likely interact to jointly produce loss spikes. Disentangling these individual contributions and accurately attributing different spike mechanisms in large-scale models remains a significant challenge for future research.

A key constraint in extending this analysis to larger models is the prohibitive computational cost of calculating Hessian eigenvalues at scale. Consequently, developing efficient algorithms to approximate the maximum eigenvalue of the Hessian and the eigenvalues in the gradient direction represents a critical direction for future work.

Furthermore, as discussed in Appendix C, the precise categorization of loss spikes into our proposed taxonomy (**neutral**, **beneficial**, **malignant**, and **catastrophic** types) presents ongoing challenges.

Developing robust, computationally efficient criteria to distinguish between these categories would significantly enhance our ability to detect and appropriately respond to different spike types during training.

B Proofs of Theoretical Results

Lemma B.1. *Let $\boldsymbol{H}$ be a real symmetric matrix and $\hat{\boldsymbol{H}} = \text{diag}\left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_t}+\varepsilon}\right)\boldsymbol{H}$. Then $\hat{\boldsymbol{H}}$ is diagonalizable in the field of real numbers.*

Proof. While $\text{diag}\left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_t}+\varepsilon}\right)\boldsymbol{H}$ is generally asymmetric, we can demonstrate that it is similar to a symmetric matrix and therefore has real eigenvalues. Let $\boldsymbol{D}_t = \text{diag}\left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_t}+\varepsilon}\right)$, which is positive definite. We can express:

$$\boldsymbol{D}_t\boldsymbol{H} = \boldsymbol{D}_t^{1/2}\cdot(\boldsymbol{D}_t^{1/2}\boldsymbol{H}\boldsymbol{D}_t^{1/2})\cdot\boldsymbol{D}_t^{-1/2}$$

Since $\boldsymbol{D}_t^{1/2}\boldsymbol{H}\boldsymbol{D}_t^{1/2}$ is symmetric, $\boldsymbol{D}_t\boldsymbol{H}$ is similar to a symmetric matrix. This confirms that $\boldsymbol{D}_t\boldsymbol{H}$ has real eigenvalues and is diagonalizable. □

Lemma B.2. *The three-term recursive iteration $\delta\boldsymbol{\theta}_{t+1} = [(1+\beta_1)\boldsymbol{I} - \eta(1-\beta_1)\boldsymbol{H}]\,\delta\boldsymbol{\theta}_t - \beta_1\delta\boldsymbol{\theta}_{t-1} - \eta(1-\beta_1)\nabla L(\boldsymbol{\theta})$ converges if and only if $\lambda_{\max}(\frac{1-\beta_1}{1+\beta_1}\boldsymbol{H}) < \frac{2}{\eta}$.*

Proof. We analyze the convergence of the vector recurrence by decomposing it along the eigenspace of the Hessian matrix. Since the Hessian $\boldsymbol{H}$ is symmetric and positive semi-definite, it admits an eigen-decomposition $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$, where $\boldsymbol{Q}$ is an orthogonal matrix and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1,\ldots,\lambda_d)$ contains the eigenvalues of $\boldsymbol{H}$.

Define the change of variables $\delta\boldsymbol{\theta}_t = \boldsymbol{Q}\boldsymbol{z}_t$. Substituting into the recurrence yields

$$\boldsymbol{z}_{t+1} = [(1+\beta_1)\boldsymbol{I} - \eta(1-\beta_1)\boldsymbol{\Lambda}]\,\boldsymbol{z}_t - \beta_1\boldsymbol{z}_{t-1} - \eta(1-\beta_1)\boldsymbol{Q}^\top\nabla L(\boldsymbol{\theta}).$$

Since this is a decoupled system in the eigenbasis, for each $i = 1,\ldots,d$, the i-th component $z_t^{(i)}$ satisfies a scalar second-order linear nonhomogeneous recurrence:

$$z_{t+1}^{(i)} = \alpha_i z_t^{(i)} - \beta_1 z_{t-1}^{(i)} + c_i,$$

where

$$\alpha_i := (1+\beta_1) - \eta(1-\beta_1)\lambda_i, \quad c_i := -\eta(1-\beta_1)g^{(i)}, \quad g^{(i)} := \left[\boldsymbol{Q}^\top\nabla L(\boldsymbol{\theta})\right]_i.$$

The general solution to this nonhomogeneous recurrence is the sum of the homogeneous solution and a particular solution. The homogeneous part is governed by the characteristic equation:

$$r^2 - \alpha_i r + \beta_1 = 0.$$

It is well known (e.g., see Elaydi, *An Introduction to Difference Equations* (Elaydi, 2005)) that the solution $z_t^{(i)}$ converges if and only if both roots of the characteristic equation lie strictly inside the unit circle in the complex plane. This is equivalent to the following three conditions:

$$\begin{aligned} 1 + \alpha_i + \beta_1 &> 0, \\ 1 - \alpha_i + \beta_1 &> 0, \\ |\beta_1| &< 1. \end{aligned}$$

Since $\beta_1 \in [0,1)$ by assumption, the third condition always holds. The first two inequalities can be rewritten as:

$$|\alpha_i| < 1 + \beta_1.$$

Substituting the expression for α_i, we obtain:

$$|(1+\beta_1) - \eta(1-\beta_1)\lambda_i| < 1 + \beta_1.$$

Solving this inequality gives:

$$0 < \eta(1-\beta_1)\lambda_i < 2(1+\beta_1) \quad \Longleftrightarrow \quad \lambda_i < \frac{2}{\eta} \cdot \frac{1+\beta_1}{1-\beta_1}.$$

Therefore, the recurrence converges in all eigendirections if and only if this condition holds for all i, i.e.,

$$\lambda_{\max}\left(\frac{1-\beta_1}{1+\beta_1}\boldsymbol{H}\right) < \frac{2}{\eta}.$$

This completes the proof. □

Theorem B.1 (Five Phases of Adam for Optimizing Quadratic Loss). *Consider the 1-d quadratic loss $L(\theta) = \frac{1}{2}\theta^2$, optimized using Adam with hyper-parameters $\beta_1 = 0$, $\beta_2 \in (0,1)$, and learning rate $\eta > 0$. The update rules are:*

$$\theta_{t+1} = \left(1 - \frac{\eta}{\sqrt{v_t}}\right)\theta_t, \quad v_{t+1} = \beta_2 v_t + (1-\beta_2)\theta_t^2.$$

Assume the initialization satisfies $v_0 = \theta_0^2$ and $|\theta_0| > \frac{\eta}{2}$. Then the training dynamics exhibit the following five-phase behavior:

(i) ***Stable Loss Decrease****. For all $t < t_0$, where*

$$t_0 := \frac{2\ln\left(\frac{|\theta_0|}{\eta} + \frac{1}{2}\right)}{\ln\frac{1}{\beta_2}},$$

the sequence $|\theta_t|$ decreases exponentially, and $v_t \in (\beta_2^t\theta_0^2, \theta_0^2)$. In particular, there exists $s \in (0,1)$ such that

$$|\theta_t| \le s^t|\theta_0|, \quad \textit{and} \quad |\theta_{t_0}| \le \delta := s^{t_0}|\theta_0|.$$

(ii) ***Decay of the Adaptive Preconditioners.*** *For $t_0 < t < t_1$, where*

$$t_1 := \inf\left\{t > t_0 \mid 1 - \frac{\eta}{\sqrt{v_t}} < -1\right\},$$

the momentum v_t decays exponentially as

$$v_t \le (v_{t_0+1} - \delta^2)\beta_2^{t-t_0-1} + \delta^2.$$

(iii) ***Onset of the Loss Spike.*** *Define*

$$t_2 := \inf\left\{t > t_1 \mid |\theta_t| > \delta\right\}.$$

For $t_1 < t < t_2$, the preconditioner v_t continues to decay, and the update multiplier $\left|1 - \frac{\eta}{\sqrt{v_t}}\right|$ grows, causing $|\theta_t|$ to increase exponentially.

(iv) ***Growth of the Adaptive Preconditioners.*** *Once $|\theta_t| > \delta$, the gradient magnitude increases, which causes v_t to grow and the update multiplier $\left|1 - \frac{\eta}{\sqrt{v_t}}\right|$ to shrink. This stabilizes the dynamics.*

(v) ***Loss Decay Phase.*** *Eventually, v_t grows large enough so that $\frac{\eta}{\sqrt{v_t}} < 1$, restoring the condition for loss decrease.*

Proof. We prove each phase sequentially.

Phase 1 (Loss Decreasing). Given $v_0 = \theta_0^2$, we first show that $v_t > \beta_2^t\theta_0^2$ by induction:

$$v_1 = \beta_2\theta_0^2 + (1-\beta_2)\theta_0^2 = \theta_0^2,$$

and for all t, since $\theta_t^2 < \theta_0^2$, we have:

$$v_{t+1} = \beta_2 v_t + (1-\beta_2)\theta_t^2 > \beta_2 v_t \Rightarrow v_t > \beta_2^t\theta_0^2.$$

This implies:

$$\frac{\eta}{\sqrt{v_t}} < \frac{\eta}{\sqrt{\beta_2^t \theta_0^2}} = \frac{\eta}{|\theta_0|} \beta_2^{-t/2}.$$

Define t_0 such that $\frac{\eta}{\sqrt{v_{t_0}}} = 1 + \frac{1}{2}$, which implies:

$$\sqrt{v_{t_0}} = \frac{\eta}{1.5} \Rightarrow v_{t_0} = \left(\frac{2\eta}{3}\right)^2.$$

Solving $\beta_2^{t_0} \theta_0^2 < v_{t_0}$, we get:

$$t_0 < \frac{\ln\left(\left(\frac{2\eta}{3}\right)^2 / \theta_0^2\right)}{\ln \beta_2} = \frac{2 \ln\left(\frac{2\eta}{3|\theta_0|}\right)}{\ln \beta_2}.$$

This shows that t_0 is finite. During this phase, we can bound the update as:

$$\theta_{t+1} = \left(1 - \frac{\eta}{\sqrt{v_t}}\right) \theta_t, \quad \text{with} \quad 0 < \frac{\eta}{\sqrt{v_t}} < 1.$$

Thus, $|\theta_t|$ decays exponentially. Let

$$s := \max\left\{\frac{1}{2}\frac{\eta}{|\theta_0|}, \left|1 - \frac{\eta}{|\theta_0|}\right|\right\} < 1,$$

then:

$$|\theta_t| \leq s^t |\theta_0|, \quad \Rightarrow \quad |\theta_{t_0}| \leq s^{t_0} |\theta_0| =: \delta.$$

Phase 2 (Decay of the Adaptive Preconditioners). For $t > t_0$, since $|\theta_t| < \delta$, we have:

$$v_{t+1} \leq \beta_2 v_t + (1 - \beta_2)\delta^2.$$

Solving the recurrence gives:

$$v_t \leq (v_{t_0+1} - \delta^2)\beta_2^{t-t_0-1} + \delta^2,$$

which shows exponential decay of v_t toward δ^2. As $v_t \to \delta^2$, the term $\frac{\eta}{\sqrt{v_t}} \to \frac{\eta}{\delta}$, which can eventually exceed 2. Therefore, there exists a finite t_1 such that:

$$1 - \frac{\eta}{\sqrt{v_{t_1}}} < -1.$$

Phase 3 (Onset of the Loss Spike). Once $1 - \frac{\eta}{\sqrt{v_t}} < -1$, the update becomes unstable:

$$\theta_{t+1} = \left(1 - \frac{\eta}{\sqrt{v_t}}\right) \theta_t, \quad \text{with} \quad \left|1 - \frac{\eta}{\sqrt{v_t}}\right| > 1.$$

Hence, $|\theta_t|$ grows exponentially. Since v_t is still small and decaying, this growth continues until $|\theta_t| > \delta$, at which point we define t_2. During this phase, v_t continues to decay, bounded as:

$$v_t \leq (v_{t_1+1} - \delta^2)\beta_2^{t-t_1-1} + \delta^2.$$

Phase 4 (Growth of the Adaptive Preconditioners). Once $|\theta_t| > \delta$, the term θ_t^2 in the update of v_t becomes significant, and v_t begins to grow. This reduces the step size $\eta/\sqrt{v_t}$, slowing down the divergence.

Phase 5 (Loss Decay Phase). Eventually, $\frac{\eta}{\sqrt{v_t}} < 1$, restoring the condition $\left|1 - \frac{\eta}{\sqrt{v_t}}\right| < 1$, and the system re-enters the stable regime where $|\theta_t|$ decreases. This completes one spike cycle. □

C Discussion: The Pros and Cons of Loss Spikes

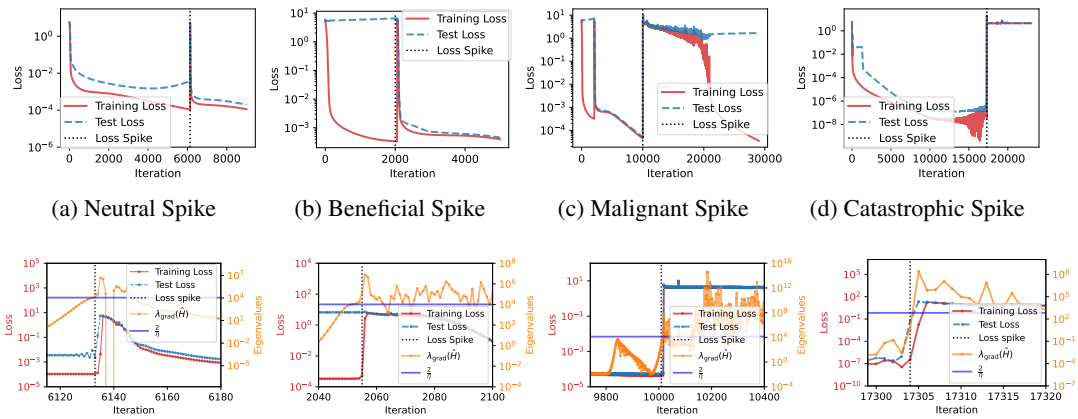
Connection to Generalization Transitions. Loss spikes represent more than mere optimization phenomena; they may signify transitions between distinct attractor basins in the optimization landscape. To systematically investigate the relationship between loss spikes and generalization, we conducted controlled experiments using a Transformer model. The model was trained to identify specific anchors within sequences, using a dataset of 2,000 samples (1,800 training, 200 test). We employed full-batch Adam optimization for training (detailed experimental setups and dataset specifications are provided in Appendix D). By analyzing the differential impacts on training and test losses before and after spike occurrences, we identified four distinct categories of loss spikes:

(i) Neutral Spikes (Fig. D1(a)): Both training and test losses resume their normal declining trajectory following the spike, suggesting minimal impact on the overall optimization process.

(ii) Beneficial Spikes (Fig. D1(b)): Prior to the spike, training loss reaches very low values while test loss remains elevated, indicating overfitting. After the spike, test loss decreases rapidly, suggesting improved generalization performance.

(iii) Malignant Spikes (Fig. D1(c)): Before the spike, both training and test losses achieve low values. After the spike, while training loss continues to decrease normally, test loss plateaus, indicating deteriorated generalization.

(iv) Catastrophic Spikes (Fig. D1(d)): Both training and test losses are low before the spike but neither recovers afterward, signifying a complete breakdown of the optimization process. These findings demonstrate that loss spikes can have context-dependent effects on generalization—sometimes enhancing model performance while in other cases degrading performance.



(a) Neutral Spike (b) Beneficial Spike (c) Malignant Spike (d) Catastrophic Spike

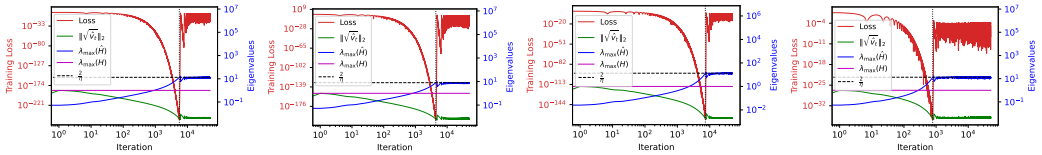
(e) Neutral Spike (f) Beneficial Spike (g) Malignant Spike (h) Catastrophic Spike

Figure D1: The Transformer model was trained to identify specific anchors within sequences. (a–d) Evolution of the training and test losses over the course of training. (e-h) Evolution of the eigenvalues in the gradient direction $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ near the spike.

As shown in Fig. D1(e–h), all four types of spikes correspond to our proposed indicator, $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$, exceeding the classical stability threshold $2/\eta$. Despite this commonality, their effects on generalization differ significantly. While our study uncovers the underlying mechanism that triggers these spikes, determining the precise conditions under which a spike becomes beneficial or malignant remains an open question for future research.

D Supplementary Experiments

Optimization of Quadratic Function with Varying Hyper-parameters. For the optimization of a one-dimensional quadratic function, Fig. D2 illustrates the precise location of the spike under various hyperparameter configurations, where $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ exceeds the stability threshold $\frac{2}{\eta}$.



(a) $\eta = 0.15, \beta_1 = 0.9, \beta_2 = 0.999$ (b) $\eta = 0.25, \beta_1 = 0.9, \beta_2 = 0.999$ (c) $\eta = 0.15, \beta_1 = 0.95, \beta_2 = 0.999$ (d) $\eta = 0.15, \beta_1 = 0.9, \beta_2 = 0.99$

Figure D2: Optimization of $f(\theta) = \frac{1}{2}\theta^2$ using the Adam algorithm with different hyperparameter settings. The solid red line denotes the training loss. The dashed black line indicates the stability threshold $\frac{2}{\eta}$. The blue, purple, and green solid lines represent $\lambda_{\max}(\boldsymbol{H}_t)$, $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$, and the bias-corrected $\|\sqrt{\hat{\boldsymbol{v}}_t}\|_2$, respectively, at each training step.

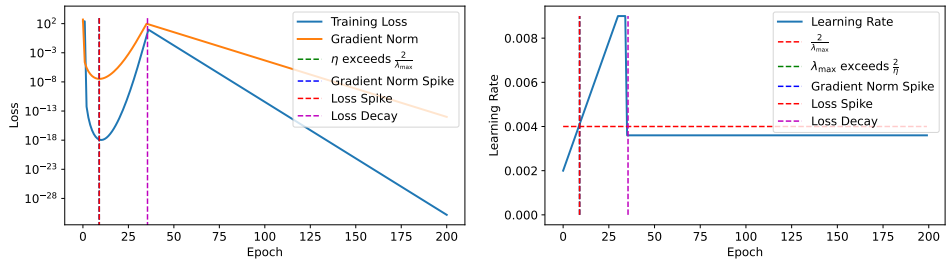
Delay Mechanism in Gradient Descent

To verify that in high-dimensional cases, when $\lambda_{\max} > \frac{2}{\eta}$, the maximum eigenvalue direction oscillates while other eigenvalue directions steadily decrease (resulting in overall loss reduction), we conducted experiments on one and two-dimensional quadratic functions with varying learning rates.

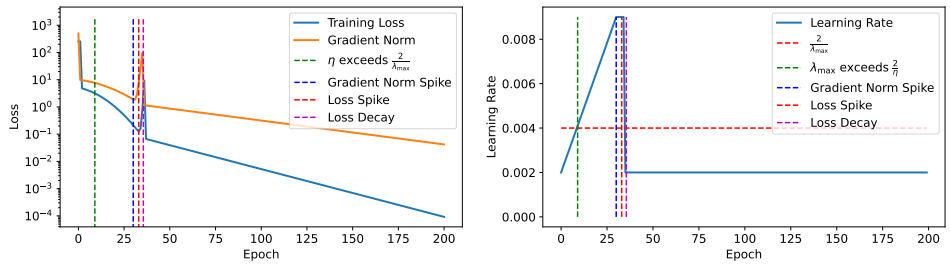
For a one-dimensional quadratic function, the loss landscape curvature remains constant. In this setting, the learning rate initially produces linear improvement over time, followed by gradual decay. When the instability condition is met—as illustrated in Fig. D3(a)—the loss increases immediately.

In contrast, for the two-dimensional case, instability primarily emerges along the dominant eigendirection, while other directions continue to descend stably. As shown in Fig. D3(b), this leads to a delayed onset of the loss spike.

To further validate this mechanism, we visualize the training trajectories in Fig. D4(a–b). In gradient descent (GD), the component along the maximum eigenvalue direction is learned rapidly at first, resulting in a small magnitude. However, once the instability condition is triggered, this component requires significant time to grow and eventually dominate the dynamics.



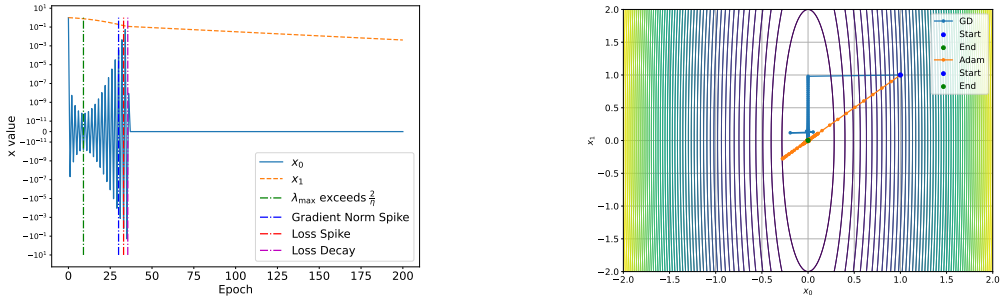
(a) 1d-quadratic $\eta = 0.15, \beta_1 = 0.9, \beta_2 = 0.99, \varepsilon = 10^{-8}$



(b) 2d-quadratic $\eta = 0.15, \beta_1 = 0.9, \beta_2 = 0.99, \varepsilon = 10^{-8}$

Figure D3: Delay mechanism in gradient descent: Comparison of loss dynamics for 1D and 2D quadratic functions. The learning rate varies over the course of training.

Gradient-direction Curvature vs. Update-direction Curvature for Loss Spike Prediction



(a) Parameter value (b) Trajectory

Figure D4: Training dynamics for the 2D quadratic function under gradient descent. (a) Evolution of the solution components along different eigendirections. (b) Optimization trajectory in parameter space.

For Adam, where the Hessian is preconditioned, we define the predictor as

$$\lambda_{\text{grad}}(\hat{\boldsymbol{H}}) := \frac{\nabla L(\boldsymbol{\theta}_t)^\top \hat{\boldsymbol{H}} \nabla L(\boldsymbol{\theta}_t)}{\|\nabla L(\boldsymbol{\theta}_t)\|^2},$$

where $\hat{\boldsymbol{H}}$ denotes the preconditioned Hessian in Eq. (7).

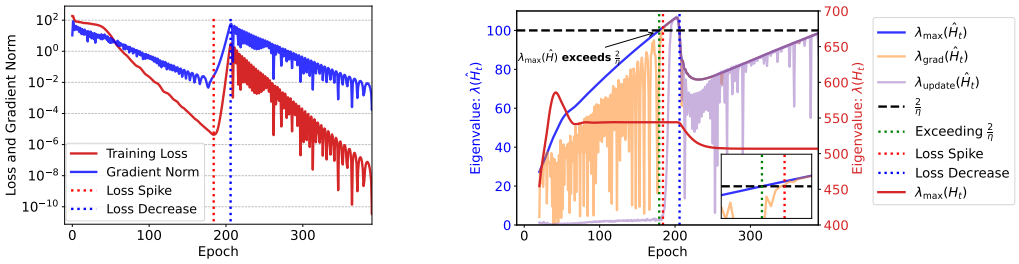
We also define

$$\lambda_{\text{update}}(\hat{\boldsymbol{H}}) := \frac{\boldsymbol{u}_t^\top \hat{\boldsymbol{H}} \boldsymbol{u}_t}{\|\boldsymbol{u}_t\|^2},$$

where $\boldsymbol{u}_t = \frac{\hat{\boldsymbol{m}}_t}{\sqrt{\hat{\boldsymbol{v}}_t}+\varepsilon}$ is the update vector.

To validate our quadratic approximation-based predictor, we tracked the eigenvalue evolution of the preconditioned Hessian throughout training. Fig. D5(b) reveals that while $\lambda_{\max}(\boldsymbol{H}_t)$ quickly stabilizes, $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ continues to increase steadily. Notably, $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ surpasses the stability threshold $\frac{2}{\eta}$ at epoch 179, yet no immediate spike occurs. At epoch 184, precisely when $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ exceeds $\frac{2}{\eta}$, we observe the loss spike depicted in Fig. D5(a). Subsequently, the eigenvalue $\lambda_{\text{update}}(\hat{\boldsymbol{H}}_t)$ in the parameter update direction also exceeds $\frac{2}{\eta}$.

This demonstrates that the eigenvalue in the gradient direction more accurately predicts the onset of the actual spike. The update direction requires time to respond to changes in the gradient. When λ_{update} exceeds $2/\eta$, the loss spike has already occurred.

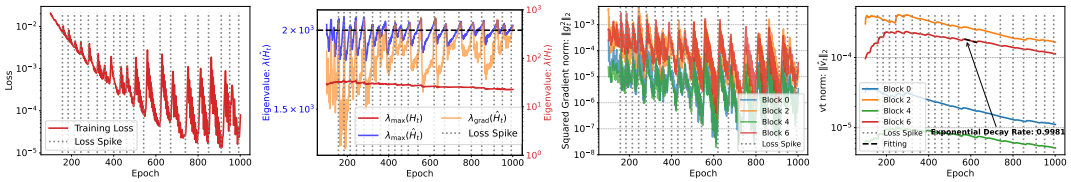


(a) Loss (b) Eigenvalues

Figure D5: (a) Training loss and gradient norm over time. (b) Evolution of critical eigenvalues: original Hessian maximum eigenvalue $\lambda_{\max}(\boldsymbol{H}_t)$, preconditioned Hessian maximum eigenvalue $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$, gradient-directional eigenvalue $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ and update-directional eigenvalue $\lambda_{\text{update}}(\hat{\boldsymbol{H}}_t)$ relative to $2/\eta$.

CIFAR-10 Experiments

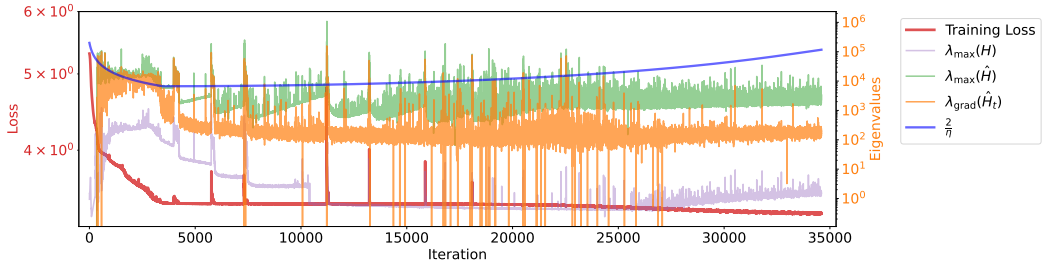
We trained a convolutional neural network on CIFAR-10 using the Adam optimizer with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The results are shown in Fig. D6. To enable efficient computation of the Hessian eigenvalues, 1,000 images were randomly selected from the CIFAR-10 dataset.



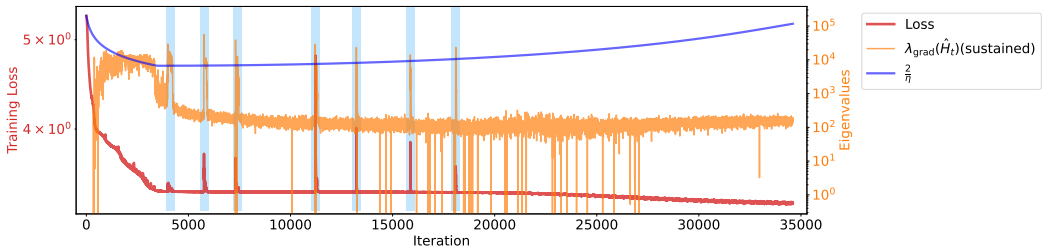
(a) Loss (b) Eigenvalues (c) Squared gradient (d) Second moment

Figure D6: Loss spike in CNNs on CIFAR10 for randomly sampled 1000 images. (a) Temporal evolution of training loss. (b) Progression of critical eigenvalue metrics: original Hessian maximum eigenvalue $\lambda_{\max}(\boldsymbol{H}_t)$, preconditioned Hessian maximum eigenvalue $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$, and gradient-directional eigenvalue $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ relative to the stability threshold $\frac{2}{\eta}$ (black dashed line). (c) Temporal evolution of gradient norm of different parameter blocks. (d) L_2-norm of second moment $\|\hat{\boldsymbol{v}}_t\|$ of different parameter blocks.

Transformer Models for Sequence Learning



(a) Eigenvalues



(b) Sustained

Figure D7: (a) Evolution of critical eigenvalues: original Hessian maximum eigenvalue $\lambda_{\max}(\boldsymbol{H}_t)$, preconditioned Hessian maximum eigenvalue $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ and gradient-directional eigenvalue $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ relative to $2/\eta$. (b) The "sustained spike predictor" evolution: $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)(\text{sustained}) = \min(\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_{t-1}), \lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t), \lambda_{\text{grad}}(\hat{\boldsymbol{H}}_{t+1}))$

For the experiment illustrated in Fig. 7, Fig. D7 presents the complete evolution of all eigenvalues, along with detailed views of each spike in Fig. 7(c-e) and Fig. D8(a-d).

As depicted in Fig. D8(a-d), we found that transient periods where $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$ and $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ exceed $2/\eta$ are insufficient to induce a spike. Loss spikes only materialize when $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$ remains above the threshold for a sustained duration. This observation aligns with stability analysis principles, which suggest that loss increases exponentially only after persistent instability, with isolated threshold violations being insufficient to trigger rapid loss elevation. Based on this insight, we formulated a

"sustained spike predictor" defined as:

$$\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)(\text{sustained}) = \min(\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_{t-1}), \lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t), \lambda_{\text{grad}}(\hat{\boldsymbol{H}}_{t+1})).$$

This refined predictor demonstrates perfect correspondence with loss spike occurrences, as shown by the orange line in Fig. D7(b).

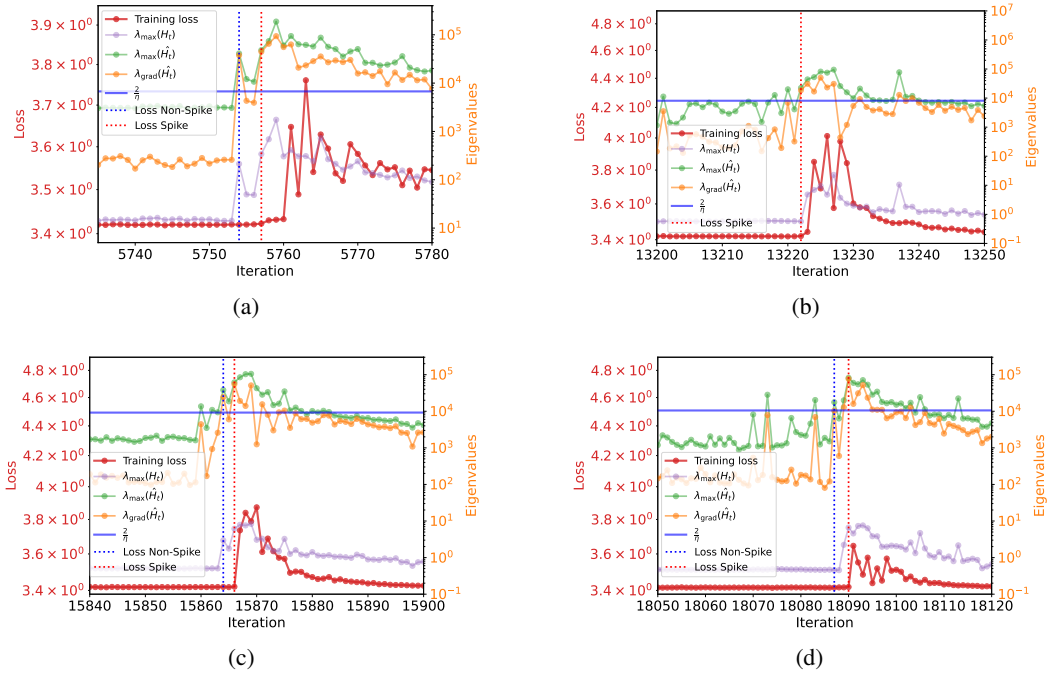


Figure D8: Detailed inspection of loss spike intervals showing the maximum eigenvalues of the original Hessian $\lambda_{\max}(\boldsymbol{H}_t)$, preconditioned Hessian $\lambda_{\max}(\hat{\boldsymbol{H}}_t)$, and $\lambda_{\text{grad}}(\hat{\boldsymbol{H}}_t)$.

Controlling Adaptive Preconditioners to Eliminate Spikes

We discovered that the epsilon parameter (ε) in Adam plays a critical role in modulating loss spike behavior. Specifically, using a larger ε significantly reduces spike severity by effectively imposing an upper bound on the preconditioned eigenvalues. Additionally, we experimented with component-wise clipping of $\boldsymbol{v}_t$, where elements falling below a specified threshold are clipped to that threshold value.

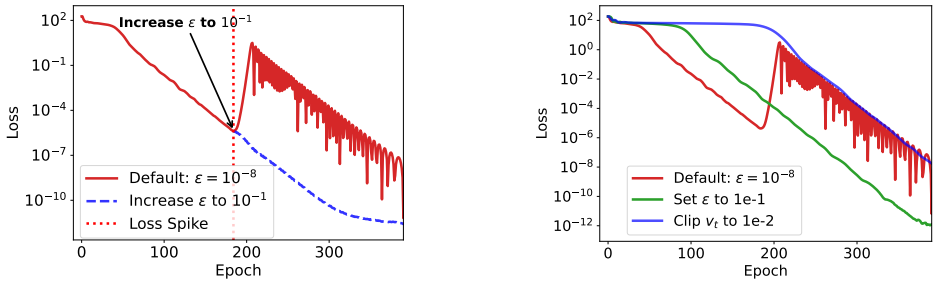


Figure D9: The training loss with the same experiment settings as Fig. 5. (a) The only difference of the orange solid line is that we change the ε in Adam to 0.1 at epoch 184 where the loss in the original training process begin to spike. (b) The orange solid line is the training loss that we change the ε to 0.1 at the beginning of the training. The blue solid line is the training loss that we clip the v_t in Adam to 0.01.

As shown in Fig. D9(a), locally increasing ε during training can effectively suppress loss spikes. Fig. D9(b) further demonstrates that increasing ε or applying $\boldsymbol{v}_t$ clipping from the beginning of training can also mitigate spike behavior, although this may come at the cost of slower convergence.

E Experimental Setup

All experiments were conducted on 1 NVIDIA RTX 4080 GPU. The runtime varied across tasks, ranging from a few minutes for smaller models to several days for large-scale training.

Computing the full Hessian matrix for large-scale neural networks is computationally prohibitive due to its quadratic memory complexity. To address this challenge, we employ an efficient power iteration method combined with Hessian-vector products that leverages automatic differentiation, circumventing the explicit construction of the complete Hessian matrix.

Setup for Fig. 4. We validate the proposed loss spike predictor using a two-layer fully connected neural network trained on 20 data points to fit the one-dimensional target function $f(x) = \sin(x) + \sin(4x)$. For panels (a)-(b), we use a hidden layer size of $m = 20$ with all parameters initialized from a Gaussian distribution ($\mu = 0$, $\sigma = m^{-0.4}$) and train using gradient descent with learning rate $\eta = 0.08$. For panels (c)-(d), we use a hidden layer size of $m = 100$ with all parameters initialized from a Gaussian distribution ($\mu = 0$, $\sigma = m^{-1}$) and train using Adam with learning rate $\eta = 0.01$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$.

Setup for Fig. 5 and Fig. 1(a). We trained two-layer fully connected neural network applied to a high-dimensional function approximation task. The target function is defined as $f^*(\boldsymbol{x}) = \boldsymbol{w}^{*\top}\boldsymbol{x} + \boldsymbol{x}^\top \mathrm{diag}(\boldsymbol{v}^*)\boldsymbol{x}$, where $\boldsymbol{w}^*, \boldsymbol{v}^* \in \mathbb{R}^{50}$ are the ground-truth parameters and $\boldsymbol{x} \in \mathbb{R}^{50}$ denotes the input features. A total of $n = 200$ data points are sampled, with inputs drawn from a standard Gaussian distribution. Gaussian noise with standard deviation $\varepsilon = 0.1$ is added to the outputs. The network has a hidden layer width of $m = 1000$, placing it in the over-parameterized regime. All weights are initialized from a Gaussian distribution $\mathcal{N}(0, \frac{1}{m})$. Training is performed using full-batch Adam with a learning rate of $\eta = 0.02$, and momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$.

Setup for Fig. 6 and Fig. 1(b). We trained a convolutional neural network on the CIFAR-10 dataset. For computational tractability in computing Hessian eigenvalues, we restricted the training set to 50 randomly sampled images. The network contains approximately $500,000$ parameters and is trained using Mean Squared Error (MSE) loss with one-hot encoded labels. Optimization is performed using full-batch Adam with a learning rate of $\eta = 0.001$ and default momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$.

Setup for Fig. 7 and Fig. 1(d). We implemented an 8-layer standard Transformer with approximately 10 million parameters. The model is trained on a synthetic dataset designed to learn compositional rules from sequences (Zhang et al., 2025), consisting of $900,000$ sequences. Training uses a batch size of 2048 and follows the next-token prediction paradigm with cross-entropy loss. The learning rate follows a linear warm-up phase followed by cosine decay. Optimization is performed using Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

Setup for Fig. D1 and Fig. 1(c). We further evaluate our theoretical insights using 4-layer and 12-layer standard Transformers trained on a synthetic classification task. The dataset is constructed to learn a specific anchor rule ($3x \to x$) from sequences (Zhang et al., 2025), comprising $2,000$ sequences. The model is trained using cross-entropy loss. The learning rate follows a linear warm-up followed by cosine decay. Adam is used for optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.