SupAno: Understanding and Suppressing Anomaly Propagation Dynamics in LoRA Fine-Tuning

Anonymous ACL submission

Abstract

Low-Rank Adaptation (LoRA) efficiently fine-tunes LLMs by freezing pretrained weights and learning low-rank increments. However, it suffers from training instability, slow convergence, and suboptimal downstream performance. Existing methods largely ignore how pretrained weights affect LoRA optimization, treating the frozen pretrained space as static. We find that anomalous weights in pretrained models, when activated by downstream data, cause an anomaly propagation dynamics consisting of three levels. It originates from block-level anomalous activations amplified by LoRA's low-rank constraint, leading to model-level forward rank collapse and backward gradient imbalance, which in turn results in a training-process-level self-reinforcing loop that drives the propagation during fine-tuning. To suppress this propagation at its source via block-level anomaly suppression, we propose a sense-decouple-suppress framework that establishes three design principles under a unified state variable: continuous sensing, deterministic decoupling, and adaptive suppression. Inspired by biological neurons, we implement the framework as SupAno, a plug-and-play anomaly suppression module at each Transformer block's output with under 0.5% parameters. Experiments on LLaMA, Mistral, Gemma, and CLIP show that SupAno preserves effective rank, mitigates gradient imbalance, and improves training stability, convergence, and downstream performance. Codes are available at https://anonymous.4open.science/r/SupAno-2B5C.

1 Introduction

Large language models (LLMs) like LLaMA, Mistral, and Gemma have advanced via self-supervised learning (Hoffmann et al., 2022; Yang et al., 2025; Guo et al., 2025). Adapting them to downstream tasks typically requires task-specific fine-tuning (Dodge et al., 2020; Ding et al., 2023; Sun et al., 2026). Low-Rank Adaptation (LoRA) freezes pretrained weights and injects trainable low-rank matrices into each layer, achieving competitive performance with high efficiency (Hu et al., 2022). However, LoRA still faces key challenges, including training instability, slow convergence, and suboptimal performance (Chang et al., 2025; Shuttleworth et al., 2026; Fan et al., 2025; Zhang et al., 2026b).

Recent LoRA advances for addressing these challenges fall into three directions.The first refines LoRA's optimization stability via improved initialization or directional decomposition within the frozen pretrained space (Hayou et al., 2024; Meng et al., 2024; Wang et al., 2024; Liu et al., 2024). The second targets dynamic rank adaptation online within the frozen base model (Zhang et al., 2023; Li et al., 2025; Valipour et al., 2023; He et al., 2025). The third stabilizes update dynamics via gradient control within the fixed pretrained landscape (Chang et al., 2025; Jiang et al., 2025). Despite their diversity, all three treat the frozen pretrained space as static, overlooking how its properties shape training dynamics, which is a critical aspect deserving further investigation.

From an optimization perspective, LoRA fine-tuning initializes from pretrained weights and continues training on downstream data. Prior work has established that the starting point of an optimization landscape governs its dynamics and the effectiveness of learning strategies (LeCun et al., 2002; Li et al., 2018; Izmailov et al., 2018). Thus, the pretrained weights, as the starting point of optimization, possess intrinsic properties that shape LoRA's training path. Unlike existing methods that adjust training strategies within a fixed space, we highlight a complementary perspective: analyzing how the intrinsic properties of pretrained weights affect LoRA optimization.

In this work, we investigate how pretrained weights affect LoRA optimization. Recent studies have revealed that pretrained weights contain struc-

tural anomalies: a small subset of weights with disproportionately large magnitudes, known as anomalous weights (Lee et al., 2024; Dettmers et al., 2024; Lin et al., 2024). During LoRA fine-tuning, these anomalous weights, when activated by downstream data, lead to a phenomenon we term **anomaly propagation dynamics**, including block-level generation and amplification, model-level propagation, and a training-process-level self-reinforcing loop. (1) *Block-level Generation and Amplification:* at the single-transformer-block level, anomalous weights produce anomalous activations, which are further amplified by LoRA's low-rank constraint. (2) *Model-level Bidirectional Propagation:* exhibiting data-driven dynamics, coupling with normal activations, and high token-level variance, these amplified block-level anomalies propagate across layers to the model level. In the forward pass, they cause effective rank collapse. In the backward pass, the collapsed rank imbalances gradients, amplifying anomalous gradients while suppressing normal ones. (3) *Training-process-level Self-Reinforcing Loop:* the imbalanced gradients further amplify the original anomalous weights during parameter updates, strengthening subsequent anomalous activations. This results in a self-reinforcing loop that drives continued propagation during fine-tuning, causing training instability and slow convergence.

To suppress this anomaly propagation dynamics at its source, we directly target block-level anomalous activations. Guided by intrinsic properties of these anomalous activations, we propose a **sense-decouple-suppress framework** that establishes three design principles: continuous sensing, deterministic decoupling, and adaptive suppression under a unified state variable. Biological neurons naturally satisfy these principles via membrane potential dynamics. Inspired by this, we design **SupAno**: a lightweight, plug-and-play anomaly suppression module inserted at each Transformer block's output that implements the framework. With under 0.5% parameter overhead, SupAno preserves effective rank, mitigates gradient imbalance, and improves training stability, convergence, and downstream performance. Our contributions are summarized as follows:

- We define anomaly propagation dynamics as the influence of pretrained anomalous weights on LoRA optimization. It consists of three levels: block-level generation and amplification, model-level bidirectional propagation causing rank collapse and gradient imbalance, and training-process-level self-reinforcing loop.
- To suppress block-level anomalous activations and suppress the propagation, we propose a sense-decouple-suppress framework that follows three design principles under a shared state variable: continuous sensing, deterministic decoupling, and adaptive suppression.
- We implement the framework as SupAno, a lightweight plug-and-play anomaly suppression module at each Transformer block's output. With under 0.5% parameter overhead, it preserves effective rank, mitigates gradient imbalance, and improves downstream performance across LLaMA, Mistral, Gemma, and CLIP.

2 Related Work

Low-Rank Adaptation Variants. Recent advances in LoRA-family methods focus on three complementary directions: convergence stability, dynamic adaptation, and training regulation. For convergence stability, DoRA (Liu et al., 2024) and LoRA+ (Hayou et al., 2024) use weight decomposition and hierarchical learning rates, while PiSSA (Meng et al., 2024) and LoRA-GA (Wang et al., 2024) further improve initialization. For dynamic adaptation, AdaLoRA (Zhang et al., 2023), TopLoRA (Li et al., 2025), and DyLoRA (Valipour et al., 2023) adjust rank online; GoRA (He et al., 2025) uses gated orthogonal transformations. For training regulation, FLoRA (Jiang et al., 2025) optimizes gradient dynamics. Despite their strengths, these methods treat pretrained weights as a fixed space and overlook how structural anomalies in pretrained weights affect LoRA optimization.

Existing Methods for Activation Processing. Existing techniques fall into two categories. The first, activation rescaling methods (e.g., SmoothQuant (Xiao et al., 2023), LLM.int8() (Dettmers et al., 2022), RMSNorm (Zhang and Sennrich, 2019)), handle large-magnitude values but do not target anomalous activations as a training-degrading phenomenon. The second, gradient modulation methods (e.g., L2 regularization (Zhang et al., 2026a), dropout (Xie et al., 2026), gradient clipping (Li et al., 2026), AdamW (Loshchilov and Hutter)), suppress anomalies but are global, indiscriminate, or stochastic. Consequently, these methods lack the ability to precisely identify or selectively suppress

anomalous activations.

Neural-inspired Adaptive Computation. Principles from biological neural systems have inspired adaptive methods for processing time-varying signals (Gast et al., 2024; Zheng et al., 2024). The Leaky Integrate-and-Fire (LIF) neuron in Spiking Neural Networks (SNNs) (Ghosh-Dastidar and Adeli, 2009; Gerstner et al., 2014) integrates sensing, gating, and adaptive response into a unified dynamical system. Recent work has incorporated such neuronal principles into deep learning for tasks with complex dynamics (Liao et al., 2024; Yan et al., 2024; Zhang et al., 2024). However, these approaches do not leverage this synergistic integration to address training instability caused by anomalous activation dynamics.

3 Dynamic Anomaly Propagation

This section analyzes how pretrained weights affect LoRA training dynamics. We first identify anomalous weights in pretrained models. When activated by downstream data, they produce block-level anomalous activations amplified by LoRA's low-rank constraint. These amplified anomalies propagate across layers, causing model-level forward rank collapse and backward gradient imbalance. Furthermore, this forms a training-process-level self-reinforcing loop driving the propagation.

3.1 Preliminary

We consider a pre-trained LLM with L layers. Fine-tuning seeks a parameter increment $\Delta\mathbf{\Theta}$ to minimize the loss on the downstream task. For a given layer ℓ, let $\mathbf{h}^{\ell-1} \in \mathbb{R}^d$ denote the input to this layer. The core linear projection operation is defined as:

$$\mathbf{z}^\ell = (\mathbf{W}_0^\ell + \Delta\mathbf{W}^\ell)\mathbf{h}^{\ell-1}, \tag{1}$$

where $\mathbf{W}_0^\ell \in \mathbb{R}^{d\times d}$ is the pre-trained weight matrix at layer ℓ, and $\Delta\mathbf{W}^\ell \in \mathbb{R}^{d\times d}$ is the learnable increment. LoRA imposes a low-rank constraint on $\Delta\mathbf{W}^\ell$ by factorizing it as $\Delta\mathbf{W}^\ell = \mathbf{B}^\ell\mathbf{A}^\ell$, where $\mathbf{B}^\ell \in \mathbb{R}^{d\times r}$, $\mathbf{A}^\ell \in \mathbb{R}^{r\times d}$, and $r \ll d$. Here, d is the hidden dimension, r is the LoRA rank, and $\mathbf{A}^\ell$, $\mathbf{B}^\ell$ are trainable matrices.

3.2 Anomalous Weight Decomposition

Prior studies have observed that pre-trained weights contain structural anomalies: a small subset of weights with disproportionately large magnitudes compared to the typical range (Lee et al., 2024; Dettmers et al., 2024; Lin et al., 2024). Building on this observation, we formally decompose the weight matrix of each layer into normal and anomalous components:

$$\mathbf{W}_0^\ell = \mathbf{W}_{\text{normal}}^\ell + \mathbf{W}_{\text{anomalous}}^\ell. \tag{2}$$

This decomposition allows us to isolate and analyze the impact of anomalous weights on LoRA's training dynamics. Although these anomalous weights are extremely few, they disproportionately influence forward computation and gradient dynamics (Lee et al., 2024; Dettmers et al., 2024), forming the basis of our analysis. In the following, we use the subscript "anom" to denote anomalous components for brevity.

3.3 Block-Level Generation and Amplification

Anomalous Activation Decomposition. Building on the decomposition of pretrained weights, the activation at layer ℓ can be expressed as the sum of three components:

$$\mathbf{z}^\ell = \underbrace{\mathbf{W}_{\text{normal}}^\ell\mathbf{h}^{\ell-1}}_{\mathbf{z}_{\text{normal}}^\ell} + \underbrace{\mathbf{W}_{\text{anom}}^\ell\mathbf{h}^{\ell-1}}_{\mathbf{z}_{\text{anom}}^\ell} + \underbrace{\Delta\mathbf{W}^\ell\mathbf{h}^{\ell-1}}_{\mathbf{z}_{\text{LoRA}}^\ell}. \tag{3}$$

Here, $\mathbf{z}_{\text{normal}}^\ell$ is the contribution from normal weights, $\mathbf{z}_{\text{anom}}^\ell$ captures the anomaly from the pretrained model, and $\mathbf{z}_{\text{LoRA}}^\ell$ is the low-rank update. Following prior work (Dettmers et al., 2022; Huang et al., 2024), we use the ℓ_2 norm as the standard metric for quantifying activation magnitudes. We compare the norms of them to identify the primary contributor to activation magnitude.

Block-Leval Generation. Since LoRA is initialized with $\Delta\mathbf{W} = \mathbf{0}$, the initialized activation is $\mathbf{z}^\ell = \mathbf{z}_{\text{normal}}^\ell + \mathbf{z}_{\text{anom}}^\ell$. To determine the dominant component, we analyze the ℓ_2 norms of $\mathbf{z}_{\text{normal}}^\ell$ and $\mathbf{z}_{\text{anom}}^\ell$. Following the spectral norm bound $\|\mathbf{W}\mathbf{h}\| \le \sigma(\mathbf{W})\|\mathbf{h}\|$ (Yoshida and Miyato, 2017; Miyato et al., 2018), we have $\|\mathbf{z}_{\text{anom}}^\ell\| \le \sigma_{\text{anom}}^\ell\|\mathbf{h}^{\ell-1}\|$ and $\|\mathbf{z}_{\text{normal}}^\ell\| \le \sigma_{\text{normal}}^\ell\|\mathbf{h}^{\ell-1}\|$. Anomalous weights in pretrained models, with magnitudes far larger than the typical range, introduce directions with abnormally large singular values (Szegedy et al., 2013), yielding $\sigma_{\text{anom}}^\ell \gg \sigma_{\text{normal}}^\ell$. We have:

$$\|\mathbf{z}_{\text{anom}}^\ell\| \gg \|\mathbf{z}_{\text{normal}}^\ell\|. \tag{4}$$

Thus, from the start of fine-tuning, each block's representation is dominated by anomalous activations generated by its pretrained anomalous weights.

Block-Leval Amplification. As established above, $\mathbf{z}^\ell_{\text{anom}}$ dominates the pretrained output. LoRA updates $\mathbf{z}^\ell_{\text{LoRA}}$ are low-rank and initialized at zero (Hu et al., 2022). Prior work shows LoRA optimization align with the top-r gradient directions of the pretrained weights (Zhang et al.). The similarity between $\mathbf{z}^\ell_{\text{LoRA}}$ and $\mathbf{z}^\ell_{\text{anom}}$ satisfies:

$$\text{sim}^\ell = \frac{\mathbf{z}^{\ell\top}_{\text{LoRA}}\mathbf{z}^\ell_{\text{anom}}}{\|\mathbf{z}^\ell_{\text{LoRA}}\|\|\mathbf{z}^\ell_{\text{anom}}\|} \geq \frac{\rho^\ell_r}{\sqrt{r}}, \tag{5}$$

where $\rho^\ell_r = \frac{\|\mathcal{P}^\ell_r(\Delta\mathbf{W}^\ell)\|_F}{\|\Delta\mathbf{W}^\ell\|_F}$, $\mathcal{P}^\ell_r$ denotes the projection onto the top-r singular subspace of the pretrained weights at layer ℓ, $\|\cdot\|_F$ is the Frobenius norm, and ρ^ℓ_r measures the fraction of LoRA's update energy aligned with this subspace. A smaller rank r increases the lower bound $\rho^\ell_r/\sqrt{r}$, forcing stronger alignment between $\mathbf{z}^\ell_{\text{LoRA}}$ and $\mathbf{z}^\ell_{\text{anom}}$:

$$\|\mathbf{z}^\ell\| = \|\mathbf{z}^\ell_{\text{anom}} + \mathbf{z}^\ell_{\text{LoRA}}\| > \|\mathbf{z}^\ell_{\text{anom}}\|. \tag{6}$$

Thus, LoRA's low-rank constraint amplifies the existing block-level anomaly, with lower r leading to stronger amplification. This constitutes the generation and amplification stage of the anomaly propagation dynamics.

3.4 Model-Level Bidirectional Propagation

Key Properties. To understand how amplified anomalies affect propagation, we characterize the intrinsic properties of block-level anomalous activations. These anomalous activations exhibit four key properties (details see Appendix A). (1) *Data-driven dynamics*: $\|\mathbf{z}^\ell_{\text{anom},i}(t)\|$ (norm of anomalous activation) varies across samples i and tokens t. (2) *Anomalous-normal coupling*: $\mathbf{z}^\ell_{\text{anom}}$ and $\mathbf{z}^\ell_{\text{normal}}$ coexist in the same matrix $\mathbf{z}^\ell$, where global operations affect both jointly. (3) *High token-level variance*: the anomaly intensity $\mathbf{z}^\ell_{\text{anom}}(t)$ varies significantly across different token positions t, with some tokens exhibiting much larger values than others. (4) *State sharing*: all properties are expressed through $\mathbf{z}(t)$, the common state variable. Driven by these properties, block-level anomalous activations propagate bidirectionally, causing forward rank collapse and backward gradient imbalance.

Forward: Effective Rank Collapse. Let $\mathbf{Z}^\ell \in \mathbb{R}^{n\times d}$ be the feature matrix stacking $\mathbf{z}^\ell$ over n samples, and let $\Sigma^\ell = \text{Cov}(\mathbf{Z}^\ell)$ be its covariance matrix. When anomalous activations dominate, the eigenvalue distribution of Σ^ℓ becomes highly skewed (Pennington and Worah, 2017), leading to a collapse in the effective rank:

$$R_{\text{eff}}(\mathbf{Z}^\ell) = \frac{\left(\sum_{i=1}^d \sigma_i(\Sigma^\ell)\right)^2}{\sum_{i=1}^d \sigma_i^2(\Sigma^\ell)} \leq C \ll d, \tag{7}$$

where $\sigma_i(\Sigma^\ell)$ are the eigenvalues of Σ^ℓ, C is an upper bound on the effective rank $R_{\text{eff}}(\mathbf{Z}^\ell)$, and d is the total feature dimension. Hence, $R_{\text{eff}} \ll d$, i.e., the block-level representation collapses to a low-dimensional subspace (see Appendix B for proof). This collapse accumulates across layers, progressively degrading model-level capacity.

Backward: Gradient Imbalance. The gradient with respect to the LoRA update is $\frac{\partial\mathcal{L}}{\partial\Delta\mathbf{W}^\ell} = \mathbf{g}^\ell(\mathbf{h}^{\ell-1})^\top$, where $\mathbf{g}^\ell = \partial\mathcal{L}/\partial\mathbf{z}^\ell$. Under Lipschitz assumptions (Glorot and Bengio, 2010), if the anomaly intensity satisfies $\|\mathbf{z}^\ell_{\text{anom}}\| \geq \xi$, then the projected gradient along the anomalous direction is bounded below:

$$\left\|\text{Proj}_{\mathbf{W}^\ell_{\text{anom}}}\left(\frac{\partial\mathcal{L}}{\partial\Delta\mathbf{W}^\ell}\right)\right\|_F \geq \alpha\xi - \beta, \tag{8}$$

where $\text{Proj}_A(B)$ denotes the projection of B onto subspace A, and $\alpha, \beta > 0$ depend on the Lipschitz constants of the loss and network. Thus, larger forward anomalies induce larger gradient updates along anomalous directions at each block (see Appendix C). This effect propagates across layers, resulting in model-level gradient imbalance.

3.5 Training-Process-Level Self-Reinforcing Loop

Gradient imbalance forces the parameter update toward the anomalous subspace at each block. The LoRA parameter update follows:

$$\Delta\mathbf{W}^\ell_{s+1} = \Delta\mathbf{W}^\ell_s - \eta_s\frac{\partial\mathcal{L}}{\partial\Delta\mathbf{W}^\ell_s}, \tag{9}$$

where η_s is the learning rate at training step s.

Due to gradient imbalance, the updated weights remain biased toward the anomalous subspace, i.e., $\|\mathbf{W}^\ell_{\text{anom}}\|$ increases while $\|\mathbf{W}^\ell_{\text{normal}}\|$ stays nearly unchanged. This directly amplifies anomalous activations in the next forward pass at the same block: $\|\mathbf{z}^\ell_{\text{anom},s+1}\| > \|\mathbf{z}^\ell_{\text{normal},s+1}\|$.

Thus, stronger anomalous activations lead to larger gradient imbalances, creating a self-reinforcing loop at the block level. As this loop operates independently across layers, its effects accumulate, leading to ill-conditioned optimization (Zhang et al.) and slow convergence (Zhuang et al., 2025) at the model level. To suppress this loop, we intervene at the source: block-level anomalous activations, motivating our method design.

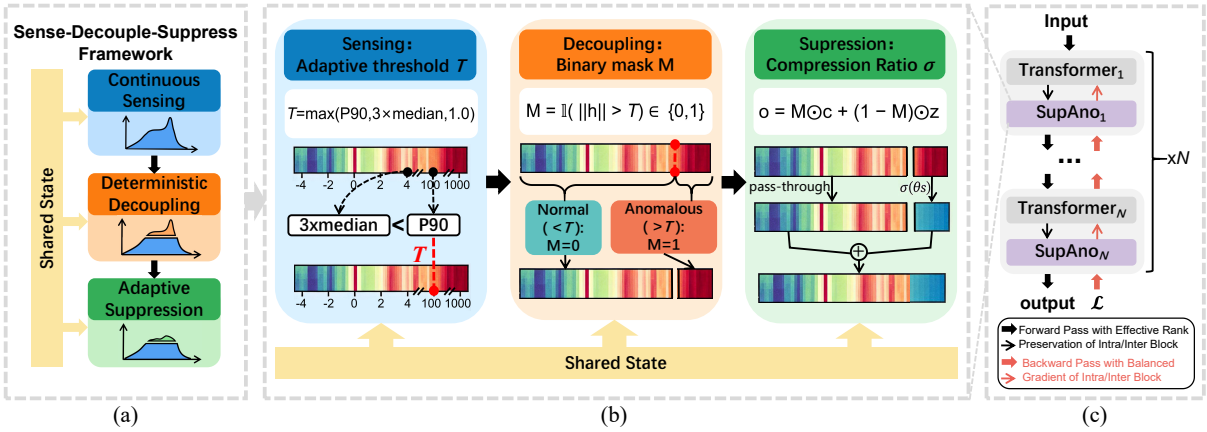

Figure 1: Overview of SupAno. (a) The Sense-Decouple-Suppress framework establishes three design principles under a shared state. (b) The SupAno module implements these principles via adaptive thresholding, binary masking, and learnable compression ratio. (c) The module is deployed as a plug-and-play component at the output of each Transformer block.

4 Method

Section 3 shows that block-level anomalous activations are the source of anomaly propagation dynamics. To suppress this propagation, this section presents our method that directly targets block-level anomalous activations. We first formalize a sense-decouple-suppress framework with three design principles under a unified state variable. Then, inspired by biological neurons, we instantiate it as SupAno, a lightweight plug-and-play anomaly suppression module at each block's output, directly addressing dynamics, coupling, and intensity variation via shared-state coordination. Figure 1 illustrates the overall architecture of our method.

4.1 Sense-Decouple-Suppress Framework

As shown in Figure 1(a), we propose a sense-decouple-suppress framework guided by the intrinsic properties of block-leval anomalous activations (dynamics, coupling, high variation, and state sharing). The framework establishes three design principles under a shared state: (1) *Continuous Sensing*: track anomaly intensity in real time to address dynamics; (2) *Deterministic Decoupling*: separate anomalies from normal activations to resolve coupling; (3) *Adaptive Suppression*: adjust suppression strength proportionally to sensed intensity to handle variation.

These three principles operate under a unified state variable, guiding our method design. Existing methods fail to satisfy these principles (see Related Work, Section 2), motivating us to seek inspiration from the Leaky Integrate-and-Fire (LIF) neuron, whose membrane potential dynamics naturally unify continuous sensing, deterministic decoupling, adaptive suppression, and state sharing.

4.2 Biological Neuron Inspiration

Based on the analysis above, we now seek a mechanism that naturally satisfies all four principles. The Leaky Integrate-and-Fire (LIF) neuron provides such a mechanism. Its membrane potential dynamics are described by:

$$\tau_m \frac{dV(t)}{dt} = -(V(t) - V_{\text{rest}}) + RI(t), \qquad (10)$$

where $V(t)$ is the membrane potential, τ_m the time constant, V_{rest} the resting potential, R the membrane resistance, and $I(t)$ the input current.

The LIF neuron instantiates the four principles as follows: (1) *Continuous sensing:* $V(t)$ integrates input $I(t)$ over time, encoding signal intensity for threshold-based decisions. (2) *Deterministic decoupling:* when $V(t) > V_{\text{th}}$, the neuron fires deterministically, separating informative spikes from noise. (3) *Adaptive suppression:* the leak term $-(V(t) - V_{\text{rest}})$ suppresses proportionally to the deviation from rest. Crucially, all three functions share the same $V(t)$: sensing accumulates, decoupling compares, and suppression resets. This shared state enables closed-loop coordination, directly motivating SupAno to implement LIF-inspired dynamics for principled anomaly suppression.

4.3 SupAno

Inspired by the LIF neuron, as illustrated in Figure 1(b), we implement the framework as SupAno,

a lightweight plug-and-play anomaly suppression module inserted at each Transformer block's output (Figure 1(c)). It operates in three stages as follows.

4.3.1 Dynamic Anomaly Sensing

To satisfy Principle 1 and inspired by the LIF neuron's input integration for state tracking and threshold-based decision, SupAno computes a per-layer adaptive threshold using a dual-benchmark strategy (Figure 1(b), left):

$$T^\ell = \max\big(P_{90}(\|\mathbf{H}^\ell\|),\ 3\times\mathrm{median}(\|\mathbf{H}^\ell\|),\ 1.0\big), \tag{11}$$

where $\mathbf{H}^\ell \in \mathbb{R}^{B\times T\times C}$ is the activation tensor, $\|\cdot\|$ the ℓ_2 norm, $P_{90}(\cdot)$ is the 90th percentile, and $\mathrm{median}(\cdot)$ denotes the median. The 90th percentile adapts to batch statistics, $3\times$ median handles isolated anomalies, and 1.0 prevents numerical issues.

4.3.2 Deterministic Decoupling

To satisfy Principle 2 and inspired by the LIF neuron's spike firing decoupling, SupAno generates a deterministic binary mask to isolate anomalous activations (Figure 1(b), middle):

$$\mathbf{M}^\ell_{b,t,i} = \mathbb{I}(\|\mathbf{h}^\ell_{b,t,i}\| > T^\ell) \in \{0,1\}, \tag{12}$$

where $\mathbf{h}^\ell_{b,t,i}$ is the activation vector and $\mathbb{I}(\cdot)$ is the indicator function, equal to 1 if $\|\mathbf{h}^\ell_{b,t,i}\| > T^\ell$ and 0 otherwise. The mask is deterministic, sparse, and detached from the computational graph to preserve gradient flow.

4.3.3 Graded Suppression

To satisfy Principle 3 and inspired by the LIF's leak-term suppression, SupAno applies dimension-wise learnable compression to anomalous activations ($\mathbf{M}^\ell = 1$) (Figure 1(b), right):

$$\mathbf{c}^\ell = \sigma(\boldsymbol{\theta}^\ell) \odot \mathbf{z}^\ell, \tag{13}$$

where $\mathbf{z}^\ell$ is the original activation, $\boldsymbol{\theta}^\ell$ learnable parameters, $\sigma(\cdot)$ the sigmoid function (output in $(0,1)$), and $\odot$ element-wise multiplication. For normal activations ($\mathbf{M}^\ell = 0$), identity preserves the signal. The final output is $\mathbf{o}^\ell = \mathbf{M}^\ell \odot \mathbf{c}^\ell + (1 - \mathbf{M}^\ell) \odot \mathbf{z}^\ell$.

All three stages share threshold T^ℓ, recomputed from output $\mathbf{o}^\ell$, forming a closed loop: sensing $\rightarrow$ decoupling $\rightarrow$ suppression $\rightarrow$ reset. By suppressing anomalies at each block, SupAno preserves effective rank and gradient balance, leading to model-level gains (Section 5).

5 Experiments

To comprehensively evaluate SupAno, we compare it with LoRA-family methods across three tasks: *(1) natural language understanding* (fine-tuning LLaMA2-7B and LLaMA3-8B on commonsense reasoning); *(2) mathematical reasoning* (fine-tuning Mistral-7B and Gemma-2 9B on MetaMathQA, evaluated on GSM8K and MATH); and *(3) image classification* (fine-tuning CLIP-ViT-B/16 on seven datasets; see Appendix E.2). Detailed experimental settings are provided in Appendix D.

5.1 Results on Language and Math Reasoning

To evaluate SupAno's generalization across reasoning domains, we conduct experiments on language understanding and mathematical reasoning. As shown in Table 1 (LLaMA2-7B; LLaMA3-8B results in Appendix E.1) and Table 2, SupAno consistently improves all LoRA-family methods. On language tasks, SupAno+LoRA achieves 78.2% average accuracy, surpassing vanilla LoRA (77.6%) by 0.6%. Gains are more significant for unstable methods: LoRA+ improves from 74.1% to 74.6%, AdaLoRA from 70.7% to 71.7%, and LoRA-SB from 81.2% to 81.5%. On challenging benchmarks such as HellaSwag and ARC-Challenge, SupAno+LoRA yields improvements of 1.8% and 0.5%, respectively.

On mathematical reasoning, SupAno consistently improves all base methods. SupAno+LoRA-Pro achieves the highest scores on Mistral-7B (GSM8K: 63.7, MATH: 17.9). For Gemma-2 9B, SupAno+DoRA attains the best MATH score (37.8), surpassing all other methods, while SupAno+LoRA-Pro achieves the highest accuracy on GSM8K (79.6). Even lightweight adapters like LoRA-SB benefit (Mistral-7B: +1.6 on GSM8K, +0.7 on MATH). SupAno acts as a plug-and-play module, requiring no architectural changes or hyperparameter tuning, demonstrating its effectiveness across diverse fine-tuning tasks. Moreover, it introduces negligible inference overhead (detailed cost analysis is deferred to Appendix F.

5.2 Comparison of Alternative Mechanisms

We compare SupAno against existing anomaly suppression techniques, including L2 regularization, Dropout, Gradient Clipping, and RMSNorm. Due to space constraints, detailed results and analysis are provided in Appendix G. In summary, SupAno

Table 1: Performance comparison of our method and LoRA-family methods on the commonsense reasoning benchmarks with LLaMA2-7B. All results are accuracy (%).

LoRA-family Methods	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
LoRA	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
SupAno+LoRA	70.2	80.4	79.8	85.4	83.2	80.2	65.2	81.4	78.2
Convergence Stability Methods									
PiSSA	67.6	78.1	78.4	76.6	78.0	75.8	60.2	75.6	73.8
LoRA+	69.4	80.1	80.6	73.4	78.3	73.7	59.3	78.2	74.1
DoRA	72.0	83.1	79.9	89.1	83.0	84.5	71.0	81.2	80.5
LoRA-SB	71.3	**85.1**	80.8	90.4	83.7	84.6	70.6	**82.8**	81.2
SupAno+LoRA+	70.3	80.5	80.3	73.8	78.6	74.2	60.1	79.0	74.6
SupAno+DoRA	**72.8**	82.8	80.3	90.5	83.7	82.8	68.3	82.4	80.5
SupAno+LoRA-SB	72.4	84.9	**81.5**	**92.4**	84.2	84.2	69.6	**82.8**	**81.5**
Dynamic Structural Adaptation Methods									
AdaLoRA	62.1	83.2	66.4	42.1	78.6	84.1	69.5	79.2	70.7
TopLoRA	70.5	83.1	81.0	87.2	83.6	83.9	68.2	82.2	80.0
SupAno+AdaLoRA	62.5	83.9	67.0	47.1	78.4	**84.5**	70.1	80.0	71.7
SupAno+TopLoRA	70.8	83.6	80.8	87.3	84.3	84.2	69.0	82.4	80.3
Training Process Regulation Methods									
Norm	69.7	83.4	78.6	87.2	81.0	81.9	66.2	79.2	78.4
SupAno+Norm	72.5	84.9	79.8	92.1	**84.7**	84.2	**71.2**	82.4	**81.5**

Table 2: Performance comparison of our method and LoRA-famliy methods on Mistral-7B and Gemma-2 9B across GSM8K and MATH.

Method	Mistral-7B		Gemma-2 9B	
	GSM8K	MATH	GSM8K	MATH
LoRA	61.9	16.0	76.2	36.6
PiSSA	62.4	16.5	77.1	37.0
DoRA	62.7	16.6	77.6	37.0
LoRA-Pro	63.1	17.3	78.3	37.5
LoRA-SB	58.9	15.3	75.4	36.7
SupAno+LoRA	62.8	17.2	78.0	37.2
SupAno+DoRA	63.4	17.3	79.2	**37.8**
SupAno+LoRA-Pro	**63.7**	**17.9**	**79.6**	37.4
SupAno+LoRA-SB	60.5	16.0	76.8	36.5

consistently outperforms these generic techniques, suggesting that its mechanism offers a reliable alternative for improving fine-tuning performance.

5.3 Component Ablation Study

To validate each component of SupAno, we conduct an ablation study on LLaMA2-7B for commonsense reasoning. Table 3 compares the SupAno+LoRA against four ablated variants: (1) SupAno-GT (global learnable threshold instead of adaptive T^ℓ), (2) SupAno-SG (soft sigmoid gate instead of hard gate $\mathbf{M}^\ell$), (3) SupAno-SC (shared scalar instead of per-dimension compression $\sigma(\boldsymbol{\theta}^\ell)$), and (4) SupAno-RM (random mask of equal sparsity instead of $\mathbf{M}^\ell = \mathbb{I}(\|\mathbf{h}^\ell\| > T^\ell)$). Full results and training dynamics are in Appendix H.

Removing any component degrades performance. Adaptive thresholding captures data-driven dynamics, contributing most significantly. Hard gating ensures deterministic decoupling and gradient stability. Per-dimension compression enables

Table 3: Ablation results on the commonsense reasoning benchmarks with LLaMA2-7B.

Method	Avg. Accuracy	Change
SupAno+LoRA	78.2%	—
SupAno-GT	76.7%	-1.5%
SupAno-SG	78.0%	-0.2%
SupAno-SC	77.8%	-0.4%
SupAno-RM	77.9%	-0.3%

precise anomaly-aware modulation. Random masking fails to achieve comparable gains, confirming that targeted anomaly suppression is critical.

5.4 Training Dynamics Analysis

To investigate SupAno's optimization mechanism, we compare LoRA and three SOTA baselines representing distinct categories: convergence stabilization (LoRA-SB), dynamic adaptation (TopLoRA), and training regulation (Norm). We evaluate their SupAno-augmented variants on LLaMA2-7B for commonsense reasoning.

Effective Rank Preservation. To assess forward representation quality, we track effective rank and activation kurtosis (He et al., 2024) across layers 5, 14, and 32 for LoRA and SupAno+LoRA (other methods in Appendix I). As shown in Figure 2, LoRA exhibits rising kurtosis and declining effective rank (a), indicating overfitting as activations sharpen while representational capacity degrades. In contrast, SupAno+LoRA (b) reverses this trend: anomalous activations are suppressed, and effective rank increases and stabilizes in a higher range (100–200). This demonstrates that SupAno mitigates distribution shift while preserving representational capacity, thereby suppressing the anomaly propagation dynamics.

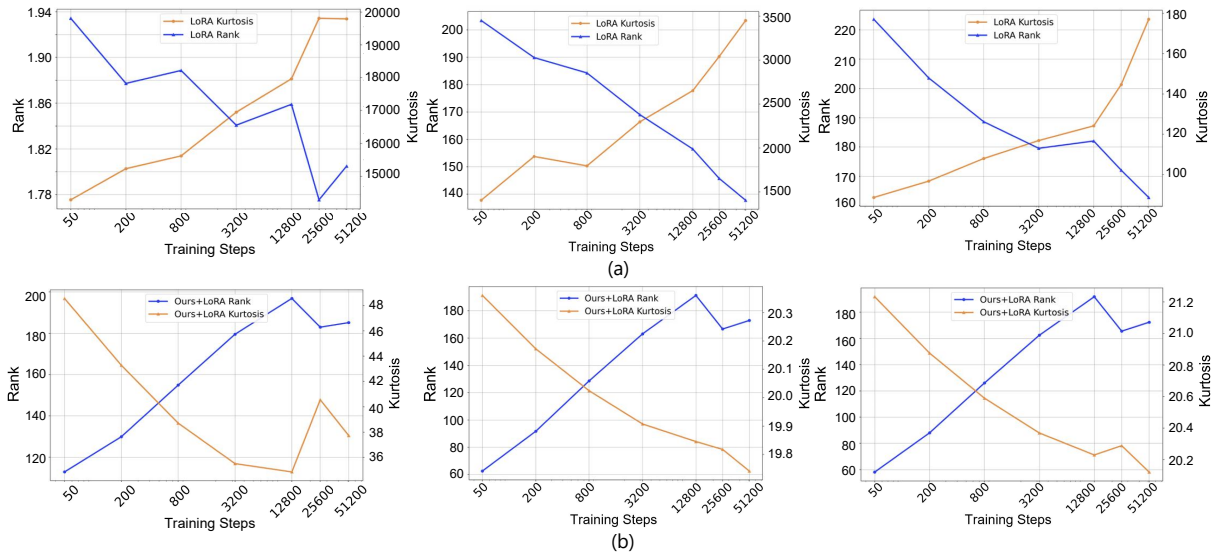

Figure 2: Evolution of activation kurtosis and effective rank in LLaMA2-7B layers 5, 14, and 32 during fine-tuning with LoRA (a) and SupAno+LoRA (b).

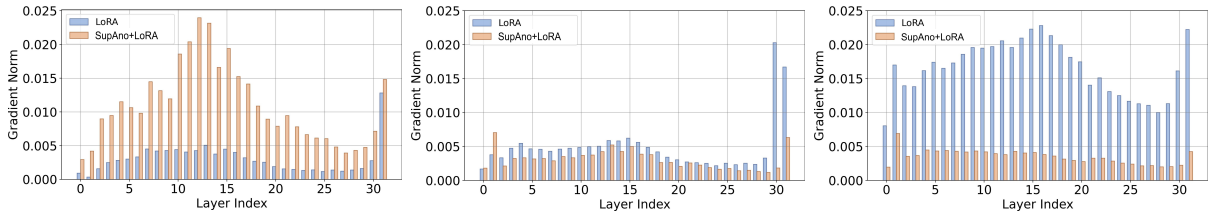

Figure 3: Comparison of gradient norms in different layers at training steps 3,200, 25,600, and 51,200 for LoRA and SupAno+LoRA.

Gradient Imbalance Mitigation. To examine SupAno's effect on gradient propagation, Figure 3 compares gradient norm evolution across layers for LoRA and SupAno+LoRA (other methods in Appendix I). LoRA shows pronounced gradient concentration in middle layers, with norms up to $4.5\times$ the layer-wise mean. SupAno mitigates this imbalance across three phases: early (3.2k steps) enhances under-activated gradients; mid (12.8k steps) achieves near-uniform distribution; late (25.6k–51.2k steps) reduces overall magnitude while preserving balance. These results confirm that SupAno effectively mitigates gradient imbalance, promoting stable and uniform optimization.

Smooth Loss Convergence. To analyze convergence behavior, Figure 4 compares loss curves of LoRA and SupAno+LoRA (other methods in Appendix I). Although SupAno+LoRA shows slightly slower initial convergence due to anomaly suppression, it subsequently achieves faster stabilization and converges to a lower final loss. These results indicate that SupAno promotes both stable and efficient convergence through its dual mechanism: suppressing forward anomalies and balancing backward gradients.

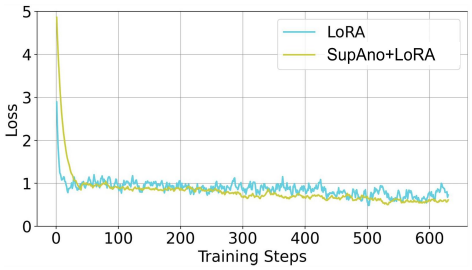

Figure 4: Training loss of LoRA and SupAno+LoRA.

6 Conclusion

This work identifies three-level anomaly propagation dynamics in LoRA fine-tuning: block-level anomalous activations cause model-level rank collapse and gradient imbalance, forming a training-process-level self-reinforcing loop. To break this cycle, we propose SupAno, a lightweight plug-and-play module inspired by biological neurons. It directly suppresses block-level anomalies while preserving normal signals, operating at each Transformer block's output with under 0.5% extra parameters. Experiments across LLaMA, Mistral, Gemma, and CLIP on multiple benchmarks show that SupAno consistently improves training stability, convergence, and downstream performance.

Limitations

While SupAno demonstrates consistent improvements across various models and benchmarks, several aspects remain for future exploration.

First, our analysis and method are primarily designed for LoRA fine-tuning. Although we have validated SupAno on multiple model families (LLaMA, Mistral, Gemma) and tasks (language understanding, mathematical reasoning), extending it to other parameter-efficient fine-tuning paradigms, such as $(IA)^3$, Adapter, or VeRA, remains an open direction. We anticipate that the core principle of anomaly suppression generalizes, but architectural differences may require task-specific adaptations.

Second, while our experiments cover language and mathematical reasoning tasks, SupAno's effectiveness in other domains (e.g., vision transformers, multimodal models, speech processing) has not yet been evaluated. Given the prevalence of anomalous activations in various neural architectures, we believe this is a promising direction for future work.

Third, SupAno introduces a small computational overhead due to per-layer threshold computation and learnable compression. Although this overhead is minimal (less than 0.5% additional parameters) and does not affect inference speed in practice, further optimization, such as fusing operations or reducing per-step recomputation, could benefit deployment in extremely resource-constrained environments (e.g., edge devices or real-time systems).

Ethic Statement

This work does not involve human subjects, crowdsourced data collection, or any form of private or sensitive data. All experiments are conducted on publicly available benchmarks and pre-trained models. We do not foresee any direct negative societal impacts of our method, as it aims to improve the training stability and efficiency of LoRA fine-tuning, which is a widely adopted technique for adapting large language models. Our code and models will be released under an open-source license to facilitate reproducibility and further research.

References

Léon Bottou, Frank E Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311.

Yupeng Chang, Chenlu Guo, Yi Chang, and Yuan Wu. 2025. Lora-mgpo: Mitigating double descent in low-rank adaptation via momentum-guided perturbation optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 648–659.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. 2017. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35:30318–30332.

Tim Dettmers, Ruslan A Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan-Adrian Alistarh. 2024. Spqr: A sparse-quantized representation for near-lossless llm weight compression. In *12th International Conference on Learning Representations*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, and 1 others. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Chenghao Fan, Zhenyi Lu, Sichen Liu, Chengfeng Gu, Xiaoye Qu, Wei Wei, and Yu Cheng. 2025. Make lora great again: Boosting lora with adaptive singular values and mixture-of-experts optimization alignment. In *International Conference on Machine Learning*, pages 15804–15832. PMLR.

Richard Gast, Sara A Solla, and Ann Kennedy. 2024. Neural heterogeneity controls computations in spiking neural networks. *Proceedings of the National Academy of Sciences*, 121(3):e2311885121.

Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. 2014. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

Samanwoy Ghosh-Dastidar and Hojjat Adeli. 2009. Spiking neural networks. *International journal of neural systems*, 19(04):295–308.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*.

Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. 2024. Understanding and minimising outlier features in transformer training. *Advances in Neural Information Processing Systems*, 37:83786–83846.

Haonan He, Peng Ye, Yuchen Ren, Yuan Yuan, Luyang Zhou, Shucun Ju, and Lei Chen. 2025. Gora: Gradient-driven adaptive low rank adaptation. *arXiv preprint arXiv:2502.12171*.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. 2013. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–8. Ieee.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 5254–5276.

Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. 2024. Rolora: Fine-tuning rotated outlier-free llms for effective weight-activation quantization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7563–7576.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

Shuyang Jiang, Yusheng Liao, Ya Zhang, Yanfeng Wang, and Yu Wang. 2025. Fine-tuning with reserved majority for noise reduction. In *The Thirteenth International Conference on Learning Representations*.

Xisen Jin, Francesco Barbieri, Brendan Kennedy, Aida Mostafazadeh Davani, Leonardo Neves, and Xiang Ren. 2021. On transferability of bias mitigation effects in language model fine-tuning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3770–3783.

Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. 2018. Adventure: Adversarial training for textual entailment with knowledge-guided examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2418–2428.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2002. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer.

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. 2024. Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13355–13364.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*.

Shiwei Li, Xiandi Luo, Haozhao Wang, Xing Tang, Ziqiang Cui, Dugang Liu, Yuhua Li, Xiuqiang He, and Ruixuan Li. 2025. Beyond higher rank: Token-wise input-output projections for efficient low-rank adaptation. *arXiv preprint arXiv:2510.23123*.

Zhiyuan Li, Yuan Wu, and Yi Chang. 2026. Aggc: Adaptive group gradient clipping for stabilizing large language model training. *arXiv preprint arXiv:2601.11864*.

Zhanfeng Liao, Yan Liu, Qian Zheng, and Gang Pan. 2024. Spiking nerf: Representing the real-world geometry by a discontinuous representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13790–13798.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, and 1 others. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 7. Granada.

Jeffrey Pennington and Pratik Worah. 2017. Nonlinear random matrix theory for deep learning. *Advances in neural information processing systems*, 30.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. 2026. Lora vs full fine-tuning: An illusion of equivalence. *Advances in Neural Information Processing Systems*, 38:174627–174662.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. 2026. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *Advances in Neural Information Processing Systems*, 38:173449–173476.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287.

Shaowen Wang, Linxi Yu, and Jian Li. 2024. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pages 38087–38099. PMLR.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE.

Sijing Xie, Dingzhu Wen, Changsheng You, Qimei Chen, Mehdi Bennis, and Kaibin Huang. 2026. Fedlodrop: Federated lora with dropout for generalized llm fine-tuning. *IEEE Journal on Selected Areas in Communications*.

Jiaqi Yan, Qianhui Liu, Malu Zhang, Lang Feng, De Ma, Haizhou Li, and Gang Pan. 2024. Efficient spiking neural network design via neural architecture search. *Neural Networks*, 173:106172.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yuichi Yoshida and Takeru Miyato. 2017. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in neural information processing systems*, 32.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Weixing Zhang, Zongrui Li, De Ma, Huajin Tang, Xudong Jiang, Qian Zheng, and Gang Pan. 2024. Spiking gs: Towards high-accuracy and low-cost surface reconstruction via spiking neuron-based gaussian splatting. *arXiv preprint arXiv:2410.07266*.

Xiang Zhang, Rui Xie, and Shikun Zhang. 2026a. L2-lora: Improving low-rank adaptation with layer-specific regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34817–34826.

Yilang Zhang, Bingcong Li, and Georgios Giannakis. 2026b. Reflora: Refactored low-rank adaptation for efficient fine-tuning of large models. *Advances in Neural Information Processing Systems*, 38:92847–92878.

Yuanhe Zhang, Fanghui Liu, and Yudong Chen. Lora-one: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *Forty-second International Conference on Machine Learning*.

Hanle Zheng, Zhong Zheng, Rui Hu, Bo Xiao, Yujie Wu, Fangwen Yu, Xue Liu, Guoqi Li, and Lei Deng. 2024. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15(1):277.

Zhan Zhuang, Xiequn Wang, Wei Li, Yulong Zhang, Qiushi Huang, Shuhao Chen, Xuehao Wang, Yanbin Wei, Yuhe Nie, Kede Ma, and 1 others. 2025. Come together, but not right now: A progressive strategy to boost low-rank adaptation. In *42nd International Conference on Machine Learning, ICML 2025*. PMLR.

A Empirical Validation of Anomalous Activation Properties

This appendix provides theoretical and empirical grounding for the four key properties of anomalous activations introduced in Section 3.4, drawing on existing literature and mathematical analysis.

Property 1: Data-driven dynamics. Prior work has shown that activation magnitudes in Transformers vary significantly across input samples and sequence positions (Kang et al., 2018; Jin et al., 2021). This input-dependent variation implies that any static threshold will be suboptimal for a subset of inputs, motivating adaptive sensing mechanisms.

Property 2: Anomalous-normal coupling. By definition, $\mathbf{z}^\ell_{\text{anom}}$ and $\mathbf{z}^\ell_{\text{normal}}$ are components of the activation vector $\mathbf{z}^\ell = \mathbf{z}_{\text{anom}} + \mathbf{z}_{\text{normal}}$. Any global operation applied to $\mathbf{z}^\ell$ (e.g., normalization, scaling) necessarily affects both components simultaneously. This inherent coupling can be quantified by the fact that $\|\mathbf{z}^\ell\|^2 = \|\mathbf{z}^\ell_{\text{anom}}\|^2 + \|\mathbf{z}^\ell_{\text{normal}}\|^2 + 2\langle \mathbf{z}^\ell_{\text{anom}}, \mathbf{z}^\ell_{\text{normal}}\rangle$, where the cross term reflects their interdependence. Thus, decoupled suppression is impossible without explicit separation.

Property 3: High token-level variance. It is well documented that anomalous activations in Transformers are highly concentrated on specific token positions, often corresponding to syntactic or semantic markers such as punctuation or separator tokens (Dettmers et al., 2022; Lee et al., 2024). This positional sparsity implies that uniform suppression strategies risk over-suppressing normal tokens or under-suppressing anomalous ones, necessitating position-aware mechanisms.

Property 4: State sharing. Properties 1–3 all concern the same activation vector $\mathbf{z}^\ell$. Property 1 describes its norm dynamics across layers and samples; Property 2 describes its decomposition into additive components; Property 3 describes its norm distribution across positions. Since all three properties reference $\mathbf{z}^\ell$, it naturally serves as the common state variable. This observation is purely structural and follows directly from the definition $\mathbf{z}^\ell = \mathbf{z}^\ell_{\text{normal}} + \mathbf{z}^\ell_{\text{anom}} + \mathbf{z}^\ell_{\text{LoRA}}$.

Summary. The four properties follow from prior empirical observations (Properties 1 and 3), algebraic decomposition (Property 2), and structural definition (Property 4). Together, they motivate the bidirectional propagation analysis in Section 3.4.

B Proof of Equation 7

B.1 Empirical Observations and Modeling

1. Let the feature matrix be $\mathbf{Z}^l \in \mathbb{R}^{n\times d}$, with covariance matrix $\Sigma^l = \text{Cov}(\mathbf{Z}^l)$.

2. To characterize the empirical behavior of LLM activations, we analyze LLaMA2/3 and Mistral-7B. For these models, the ℓ_2-aggregated energy of singular values concentrates on a low-dimensional subspace S with $|S| = s \approx 3$–10. The concentration coefficient κ, defined below, is consistently observed in the range $\kappa \in [1.5, 2.5]$:

$$\sqrt{\sum_{i\in S}\sigma_i^2} \;\geq\; \kappa\sqrt{\sum_{i\notin S}\sigma_i^2}, \quad |S| = s \ll d,\ \kappa > 1, \tag{A.1}$$

   where σ_i are the singular values of Σ^l.

3. The effective rank is defined as (Pennington and Worah, 2017):

$$R_{\text{eff}}(\mathbf{z}^l) = \frac{(\sum_{i=1}^d \sigma_i)^2}{\sum_{i=1}^d \sigma_i^2}. \tag{A.2}$$

B.2 Proof for the Single-Direction Case ($s = 1$)

We first consider the simplified case where the ℓ_2 energy concentrates on a single direction ($s = 1$),

i.e., $\sigma_1 \geq \kappa \sum_{i=2}^{d} \sigma_i$. This provides a building block for the general subspace case.

Step 1: Singular value relation.

$$\sigma_1 \geq \kappa \sum_{i=2}^{d} \sigma_i. \tag{A.3}$$

Let $S = \sum_{i=2}^{d} \sigma_i$, then $\sigma_1 \geq \kappa S$.

Step 2: Numerator bound.

$$\sum_{i=1}^{d} \sigma_i = \sigma_1 + S \geq (\kappa + 1)S. \tag{A.4}$$

Step 3: Denominator lower bound. By Cauchy–Schwarz, $S^2 \leq (d-1)\sum_{i=2}^{d} \sigma_i^2$, so $\sum_{i=2}^{d} \sigma_i^2 \geq S^2/(d-1)$. Thus:

$$\sum_{i=1}^{d} \sigma_i^2 \geq \sigma_1^2 + \frac{S^2}{d-1} \geq \kappa^2 S^2 + \frac{S^2}{d-1} \tag{A.5}$$

$$= S^2 \left(\kappa^2 + \frac{1}{d-1} \right). \tag{A.6}$$

Step 4: Effective rank bound.

$$R_{\text{eff}} \leq \frac{[(\kappa+1)S]^2}{S^2 \left(\kappa^2 + \frac{1}{d-1} \right)} = \frac{(\kappa+1)^2}{\kappa^2 + \frac{1}{d-1}}. \tag{A.7}$$

Step 5: Asymptotic bound. As $d \to \infty$, $R_{\text{eff}} \leq \frac{(\kappa+1)^2}{\kappa^2} = 1 + \frac{2}{\kappa} + \frac{1}{\kappa^2} \stackrel{\text{def}}{=} C$. Thus R_{eff} is bounded by a dimension-independent constant.

B.3 Extension to the Subspace Case ($s \approx 3$–10)

Empirical observations from LLaMA2/3 and Mistral-7B show that the spectral energy is distributed across a low-dimensional subspace of size $s \approx 3$–10, rather than a single direction. The largest singular value contributes only 17%–28% of the total energy, while the top s singular values collectively account for 45%–65%. The concentration coefficient κ falls in $[1.5, 2.5]$ as defined above.

Following the same analytical framework as the single-direction case, the effective rank satisfies:

$$R_{\text{eff}} \leq \left(\sqrt{s} + \frac{\sqrt{r}}{\kappa} \right)^2, \quad r = d - s. \tag{A.8}$$

Derivation sketch. Let $A = \sum_{i \in S} \sigma_i$, $B = \sum_{i \notin S} \sigma_i$, $A_2 = \sum_{i \in S} \sigma_i^2$, $B_2 = \sum_{i \notin S} \sigma_i^2$. By Cauchy–Schwarz, $A \leq \sqrt{sA_2}$, $B \leq \sqrt{rB_2}$. From $\sqrt{A_2} \geq \kappa\sqrt{B_2}$ we have $\sqrt{B_2} \leq \sqrt{A_2}/\kappa$. Thus $A + B \leq \sqrt{A_2}(\sqrt{s} + \sqrt{r}/\kappa)$. Since $A_2 + B_2 \geq A_2$, the bound follows.

Illustrative example (LLaMA2-7B). For typical values $s = 5$, $\kappa = 2$, $d = 4096$:

$$R_{\text{eff}} \leq (\sqrt{5} + 32)^2 \approx 1172 \ll 4096. \tag{A.9}$$

Remark. This bound captures the qualitative effect that the effective rank is controlled by the subspace size s and concentration κ, rather than the full dimension d.

C Proof of Equation 8

C.1 Definitions and Setup

We establish the following setup for the analysis:

1. The loss function $\mathcal{L}$ satisfies standard Lipschitz continuity and smoothness assumptions, which is standard in optimization theory for analyzing gradient dynamics (Bottou et al., 2018) and commonly adopted in fine-tuning studies (Hayou et al., 2024).

2. The forward pass produces an anomalous activation $\mathbf{z}^\ell_{\text{anom}}$ satisfying $\|\mathbf{z}^\ell_{\text{anom}}\| \geq \xi$, where $\xi > 0$ is a lower bound on the anomaly intensity.

3. The gradient with respect to the LoRA update at layer ℓ is given by the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \Delta \mathbf{W}^\ell} = \mathbf{g}^\ell (\mathbf{h}^{\ell-1})^\top, \quad \text{where } \mathbf{g}^\ell = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^\ell}. \tag{A.10}$$

C.2 Proof Sketch

The following derivation provides an illustrative bound intended to capture the qualitative direction of the effect: larger forward anomalies induce larger gradient updates along anomalous directions. Constants $\alpha, \beta > 0$ absorb task-specific, architecture-dependent factors.

Step 1: Gradient decomposition. The activation $\mathbf{z}^\ell$ decomposes into normal and anomalous components:

$$\mathbf{z}^\ell = \mathbf{z}^\ell_{\text{normal}} + \mathbf{z}^\ell_{\text{anom}}, \tag{A.11}$$

where $\|\mathbf{z}^\ell_{\text{anom}}\| \geq \xi$, $\|\mathbf{z}^\ell_{\text{anom}}\| \gg \|\mathbf{z}^\ell_{\text{normal}}\|$. Since $\mathbf{z}^\ell_{\text{anom}}$ dominates, the gradient $\mathbf{g}^\ell = \partial\mathcal{L}/\partial\mathbf{z}^\ell$ has a

significant component along the direction of $\mathbf{z}^{\ell}_{\text{anom}}$. Let $\mathbf{u}$ be the unit vector in that direction. By Lipschitz smoothness:

$$\left\langle \mathbf{g}^{\ell}, \mathbf{u} \right\rangle \geq \gamma\xi, \tag{A.12}$$

where $\gamma > 0$ is a constant depending on the curvature of the loss.

Step 2: Projection onto the anomalous subspace. Let $\text{Proj}_{\mathbf{W}^{\ell}_{\text{anom}}}(\cdot)$ denote projection onto the subspace spanned by the anomalous weights. Since $\mathbf{z}^{\ell}_{\text{anom}}$ is produced by $\mathbf{W}^{\ell}_{\text{anom}}$ acting on the input $\mathbf{h}^{\ell-1}$, the component of $\mathbf{g}^{\ell}$ along $\mathbf{z}^{\ell}_{\text{anom}}$ contributes directly to the update in the anomalous subspace. The Frobenius norm of the projected gradient satisfies:

$$\left\| \text{Proj}_{\mathbf{W}^{\ell}_{\text{anom}}} \left(\frac{\partial \mathcal{L}}{\partial \Delta \mathbf{W}^{\ell}} \right) \right\|_F \geq \alpha \|\mathbf{z}^{\ell}_{\text{anom}}\| - \beta, \tag{A.13}$$

where $\alpha, \beta > 0$ absorb constants from the Lipschitz constants of the loss, the network, and the alignment between $\mathbf{g}^{\ell}$ and $\mathbf{z}^{\ell}_{\text{anom}}$.

Step 3: Interpretation. The inequality shows that the magnitude of the gradient update along the anomalous direction grows at least linearly with the anomaly intensity $\|\mathbf{z}^{\ell}_{\text{anom}}\|$. Consequently, larger forward anomalies induce larger gradient updates along the anomalous subspace, which reinforces the original anomalous weights during parameter updates.

C.3 Conclusion

The derivation above establishes that anomalous activations lead to disproportionately large gradient updates in the anomalous weight direction. This mechanism creates a positive feedback loop: stronger anomalies $\rightarrow$ larger imbalanced gradients $\rightarrow$ further amplification of anomalous weights $\rightarrow$ even stronger anomalies in subsequent forward passes. The constants α and β depend on task-specific and architecture-specific factors, but the linear dependency $\alpha\|\mathbf{z}^{\ell}_{\text{anom}}\|$ guarantees that the amplification effect is inevitable when anomalies are present.

D Experimental Settings

Evaluation on Language Understanding Tasks. Following the setup of Hu et al.(Hu et al., 2023), we combine the training sets from all eight tasks into a single training corpus, while evaluation is performed separately on the held-out test set of each individual task. LoRA modules are inserted into the key, value, query, attention-output, and all feed-forward weight matrices, using a rank of $r = 32$.

Evaluation on Mathematical Reasoning Tasks. Experiments employ a unified LoRA configuration, where adapters are applied to the query, key, value, and output weight matrices of attention modules, as well as to all feed-forward layers, with rank values $r = 32$.

Environment. All experiments were conducted on NVIDIA A100 80GB GPUs with CUDA 12.1 and PyTorch 2.1.0.

E Experimental Results

E.1 Commonsense reasoning tasks on LLaMA3-8B.

We fine-tune LLaMA3-8B on a collection of commonsense reasoning tasks comprising eight subtasks, each with its own predefined training and test splits. The results are summarized in Table 4.

SupAno consistently improves the performance of all LoRA-family methods it is combined with. When integrated with standard LoRA, SupAno+LoRA achieves an average accuracy of 82.2%, outperforming vanilla LoRA (80.8%) by 1.4%. For the relatively weaker baseline PiSSA, SupAno+PiSSA achieves 78.0%, a substantial gain of 2.6% over PiSSA (75.4%), demonstrating SupAno's effectiveness in stabilizing training for methods more susceptible to anomalous activations.

Notably, SupAno also delivers consistent improvements over strong baselines and on challenging benchmarks. SupAno+DoRA attains an average accuracy of 86.1%, slightly outperforming standalone DoRA (85.2%) by 0.9%, with particularly notable improvements on HellaSwag (96.3% vs. 95.5%) and OBQA (87.5% vs. 85.8%). Similarly, SupAno+TopLoRA achieves a new state-of-the-art average of 87.6%, surpassing the already strong TopLoRA baseline (86.9%) by 0.7%, while maintaining competitive performance on ARC-c (94.8%) and HellaSwag (95.5%). The results further highlight SupAno's plug-and-play nature: it can be seamlessly integrated with diverse LoRA variants, including standard LoRA, PiSSA, DoRA, and TopLoRA—without requiring method-specific adjustments, delivering consistent performance gains across the board.

Table 4: Performance comparison of our method and LoRA-family methods on the commonsense reasoning benchmarks with LLaMA3-8B. All results are accuracy (%).

LoRA-family Methods	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
LoRA	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
PiSSA	67.1	81.1	77.2	83.6	78.9	77.7	63.2	74.6	75.4
DoRA	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	85.2
TopLoRA	74.3	90.1	83.4	94.5	88.0	82.0	94.0	88.4	86.9
SupAno+LoRA	72.0	86.2	81.0	93.5	85.5	85.8	73.0	80.5	82.2
SupAno+PiSSA	69.5	83.0	79.0	87.0	81.0	80.0	66.5	77.5	78.0
SupAno+DoRA	**75.5**	90.0	81.0	**96.3**	86.5	**91.0**	81.0	87.5	86.1
SupAno+TopLoRA	75.0	**90.8**	**84.2**	95.5	**88.8**	82.7	**94.8**	**89.0**	**87.6**

Table 5: Performance of fine-tuning CLIP-VIP-B/16 on 7 image classification tasks.

LoRA-family Methods	Cars	DTD	EuroSAT	GTSRB	RESISC45	SUN397	SVHN	Avg.
LoRA	72.8	73.9	96.9	92.4	90.0	70.1	88.0	83.4
LoRA+	72.9	74.1	97.0	92.4	90.0	70.2	88.1	83.5
DoRA	73.7	73.7	97.0	92.4	90.0	70.2	88.2	83.6
LoRA-GA	85.2	77.5	98.0	95.3	94.4	75.4	94.6	88.6
SupAno+LoRA	73.4	74.3	97.2	93.5	90.4	70.3	88.4	83.9
SupAno+LoRA+	72.5	74.3	97.6	92.6	82.7	90.3	70.5	82.9
SupAno+DoRA	73.4	73.9	97.4	92.2	90.5	70.6	88.8	83.8
SupAno+LoRA-GA	**85.5**	**77.9**	**98.3**	**95.7**	94.2	75.6	**95.0**	**88.9**

Compared with the results on LLaMA2-7B, LLaMA3-8B exhibits more pronounced performance gains when enhanced with SupAno. For instance, SupAno+LoRA yields a 1.4% improvement on LLaMA3-8B, compared to only 0.6% on LLaMA2-7B; SupAno+TopLoRA achieves a 0.7% gain on LLaMA3-8B, versus 0.3% on LLaMA2-7B. This observation validates that despite more extensive pre-training, the increased parameter count in larger models (8B vs. 7B) introduces more complex anomalous activation distributions, allowing SupAno's dynamic anomaly suppression mechanism to exert a greater effect.

E.2 Evaluation on Image Classification Tasks

We evaluate the performance of SupAno-enhanced LoRA-family methods on eight image classification benchmarks by comparing our approach with several state-of-the-art baselines. Experiments are conducted by fine-tuning the CLIP-ViT-B/16 visual backbone (Radford et al., 2021) on datasets including StanfordCars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Houben et al., 2013), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2010), and SVHN (Netzer et al., 2011). The text classifier remains frozen using prompt templates such as "a photo of a {class}." Accuracy serves as the evaluation metric.

The results are summarized in Table 5. SupAno consistently improves the base LoRA-family methods across almost all datasets. Notably, SupAno+LoRA-GA achieves the highest average accuracy of 88.9%, surpassing the strong baseline LoRA-GA (88.6%) by 0.3% and outperforming standard LoRA (83.4%) by 5.5%. Moreover, SupAno+LoRA achieves 83.9%, exceeding vanilla LoRA (83.4%) by 0.5% and slightly outperforming LoRA+ (83.5%) and DoRA (83.6%). These gains are observed even in challenging fine-grained recognition tasks such as StanfordCars and RESISC45, where SupAno+LoRA-GA reaches 85.5% and 94.2%, respectively. The consistent improvements across diverse architectures, including LoRA, DoRA, and LoRA-GA, demonstrate that SupAno serves as a lightweight, plug-in module that enhances training stability and generalization without introducing substantial computational overhead.

Table 6: Inference latency comparison on NVIDIA A100 (LLaMA-7B, batch=1, sequence length=512).

Configuration	Latency (ms)	Overhead
LoRA (merged)	baseline (effectively 0)	—
LoRA (unmerged)	45.2	reference
SupAno+LoRA	**46.4**	**+2.7%**

F Inference Efficiency Analysis

To quantify the inference latency overhead introduced by SupAno's input-dependent dynamic gat-

Table 7: Comparison of SupAno with general regularization methods on LLaMA2-7B commonsense reasoning (average accuracy %).

Method	Avg. Accuracy	vs. LoRA
LoRA (baseline)	77.6	—
L2 Regularization	77.9	+0.3
LR Decay (0.50x)	77.8	+0.2
LR Decay (0.25x)	77.1	-0.5
Dropout (0.1)	77.3	-0.3
Gradient Clipping	78.0	+0.4
SupAno+LoRA	**78.2**	**+0.6**

ing, we conduct experiments on NVIDIA A100 using LLaMA2-7B with batch size 1 and sequence length 512. Each measurement is repeated 100 times, and the average latency is reported.

As shown in Table 6, the LoRA baseline incurs 45.2 ms latency. SupAno adds approximately 1.2 ms, resulting in 46.4 ms total latency, a relative overhead of 2.7%. This modest cost is substantially outweighed by the performance gains (0.5%–3.1% accuracy improvement across tasks) and training stability benefits. Additionally, SupAno introduces less than 0.5% additional parameters and incurs only 3%–5% training time overhead.

G Comparison of Alternative Mechanisms

We compare SupAno against existing anomaly suppression techniques: regularization, dropout, learning rate decay, and gradient clipping. All experiments use LLaMA2-7B on commonsense reasoning benchmarks with LoRA as the base fine-tuning method. For L2 regularization, we search coefficient in $\{0.01, 0.05, 0.1\}$ and report the best. Learning rate decay uses $0.50\times$ and $0.25\times$ of base rate (1×10^{-4}). Dropout (0.1) is applied to hidden activations. Gradient clipping uses norm threshold 1.0. SupAno follows default configuration with $< 0.5\%$ parameter overhead.

As shown in Table 7, reported results demonstrate that SupAno achieves the highest accuracy among all methods (78.2%, +0.6%). Gradient clipping (+0.4%) and L2 regularization (+0.3%) yield smaller gains, while dropout (-0.3%) and learning rate decay (-0.5%) degrade performance. These results confirm that SupAno consistently outperforms other generic techniques, suggesting that its targeted anomalous activations suppression mechanism offers a reliable alternative for improving fine-tuning performance.

H Component Ablation Study

To validate each component of SupAno, we conduct an ablation study on LLaMA2-7B for commonsense reasoning, with SupAno+LoRA as the baseline. Table 3 reports final accuracy, and Figs. 5–7 show the evolution of training dynamics.

Adaptive vs. Global Thresholding. To evaluate the necessity of layer-wise adaptive thresholding, we compare SupAno against a variant with a global learnable threshold. Figure 5 shows that SupAno's threshold closely tracks per-layer dynamics, whereas the global threshold does not, leading to persistent anomalies and rank decay. In contrast, full SupAno maintains stable rank and effective anomaly control.

Hard vs. Soft Gating. To examine the impact of gating style on gradient stability, we replace the hard binary gate with a soft sigmoid gate, which introduces indiscriminate modulation and noisy gradients. Figure 6 compares gradient-norm distributions across layers. Hard gating yields stable, low-variance distributions, preserving gradient balance. In contrast, soft gating produces distributions with greater inter-layer spread and higher sensitivity to training steps, as reflected by its significantly larger coefficient of variation. These results confirm that deterministic, sparse anomaly isolation is essential for gradient stability and balanced optimization.

Per-Dimension vs. Shared Compression. To evaluate per-dimension adaptive compression, we compare SupAno against a variant with globally shared compression in Figure 7, analyzing top-5% anomalous activations and their corresponding compression strengths at three training stages (3,200, 25,600, and 51,200 steps). Shared compression applies uniform scaling across all stages, failing to adapt to evolving anomaly patterns. In contrast, SupAno adaptively adjusts compression strength per dimension over time, progressively reducing anomalous activations while preserving normal signals. This trend demonstrates that SupAno's per-dimension adaptive compression enables precise, stage-aware anomaly modulation and more stable optimization.

Random vs. Targeted Masking. To verify that SupAno's gains stem from targeted anomaly suppression rather than random regularization, we compare against two variants: one with a random mask of equal sparsity (Random Mask + Compression), and another using only random masking without com-

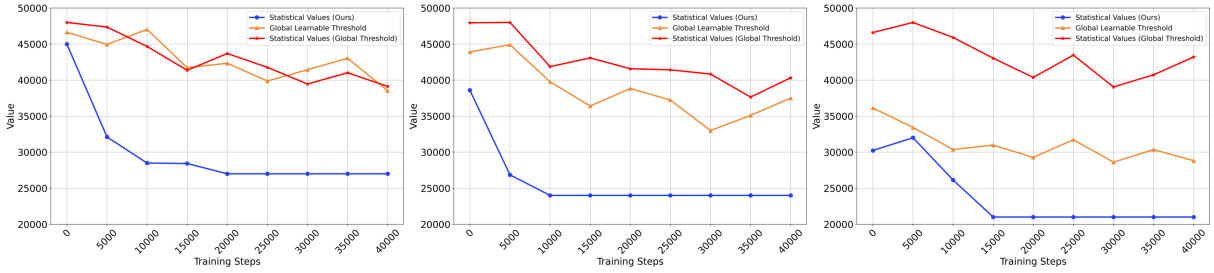

Figure 5: Comparison between SupAno's adaptive threshold and a globally learnable threshold in tracking the 90th percentile of activations throughout training, shown for layers 1, 16, and 32.

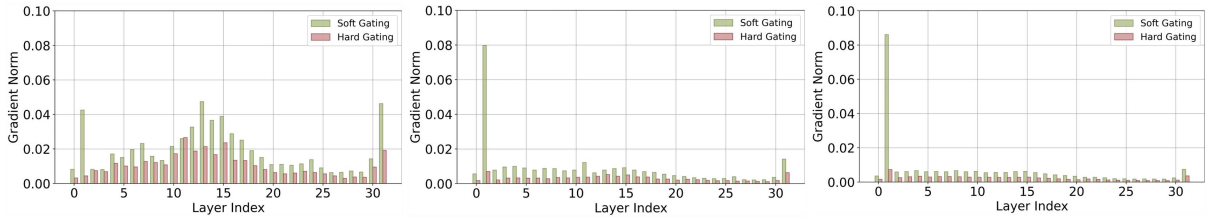

Figure 6: Comparison of gradient norm distributions across layers at training steps 3,200, 25,600, and 51,200 under hard and soft gating.

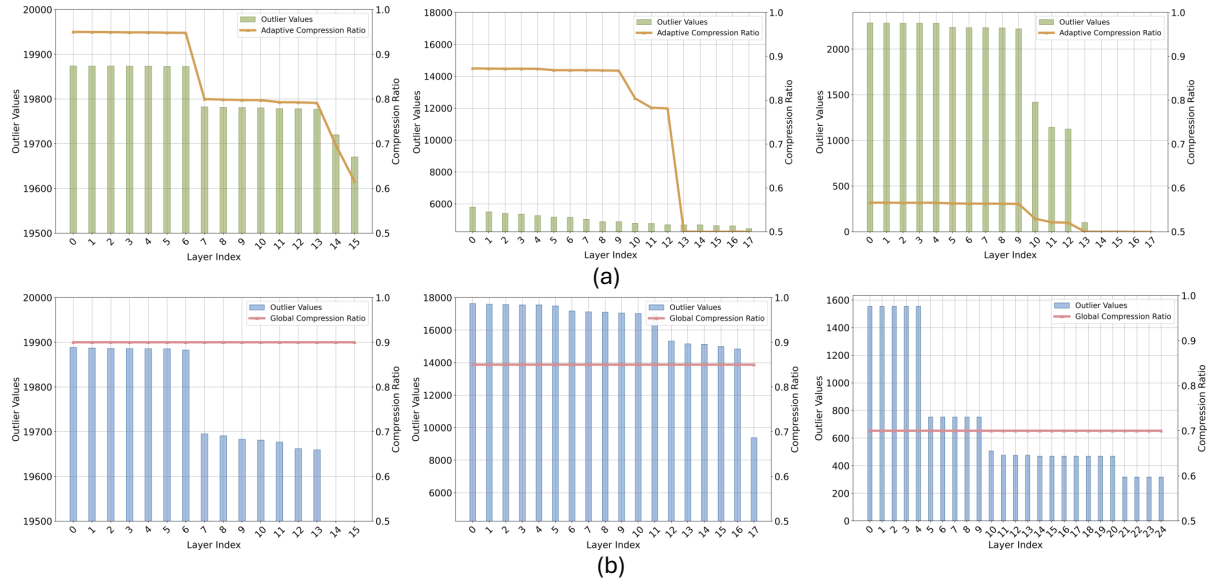

Figure 7: Comparison of top-5% anomalies and their corresponding compression strengths under SupAno (a) and a globally-shared compression parameter (b) at three training stages (3,200, 25,600, and 51,200 steps).

pression (Pure Random Masking). As shown in Table 8, Random Mask + Compression achieves only 77.9%, far below SupAno's 78.2%, while Pure Random Masking drops further to 73.1% on Stanford Cars. These results confirm that the sparsity pattern, specifically targeting actual anomalies, is critical. Random dropping with the same density provides marginal benefit, whereas SupAno's anomaly-aware targeting reverses rank collapse and achieves substantial gains.

Table 8: Targeted suppression validation. Results: Avg. Accuracy (8-task commonsense) / Stanford Cars (long-tail).

Method	Commonsense Avg.	Stanford Cars
Global Threshold	77.9	80.1
Random Mask + Compression	77.9	78.3
Pure Random Masking	77.4	73.1
SupAno (full)	**78.2**	**85.5**

I Supplementary Results for Training Dynamics Analysis

Due to space constraints, we present the comparative results of training dynamics, including

the evolution of rank, layer-wise gradient norm distributions, and loss curves, for Norm vs. SupAno+Norm, TopLoRA vs. SupAno+TopLoRA, and LoRA-SB vs. SupAno+LoRA-SB in Figures 8–12. Across all three base methods, SupAno consistently improves training dynamics.

Effective Rank preservation. SupAno+Norm maintains higher effective rank throughout training compared to Norm, which exhibits steady rank decay. Similarly, SupAno+TopLoRA and SupAno+LoRA-SB show rank stabilization at significantly higher levels than their base counterparts.

Gradient Imbalance Mitigation. As shown in the layer-wise gradient norm figures, SupAno reduces inter-layer variance and prevents gradient concentration in middle layers. For Norm, gradients are heavily concentrated in layers 10–20, with norms up to $4\times$ the mean. SupAno+Norm flattens this distribution, achieving near-uniform gradient norms across layers. Similar patterns are observed for TopLoRA and LoRA-SB.

Smooth Loss Convergence. Although SupAno variants exhibit slightly slower initial descent due to threshold adaptation, they achieve faster stabilization and lower final loss values after a brief warm-up phase ($\sim$50 steps). For example, SupAno+LoRA-SB converges to a final loss of 0.2, compared to 0.6 for LoRA-SB.

These results collectively confirm that SupAno's sense-decouple-suppress mechanism generalizes across different LoRA variants, preserving effective rank, balancing gradient propagation, and improving convergence stability.

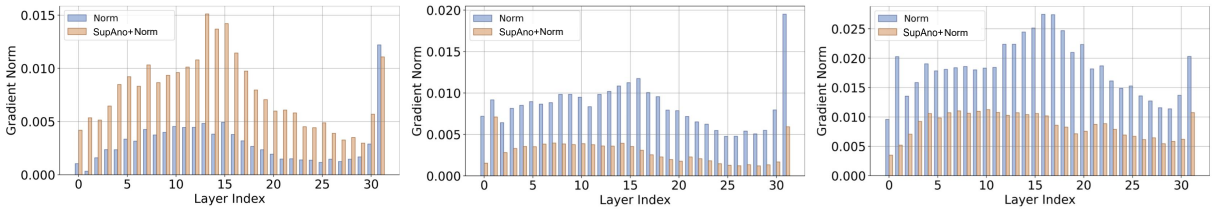

Figure 8: Comparison of gradient norms in different layers at training steps 3,200, 25,600, and 51,200 for Norm and SupAno+Norm.

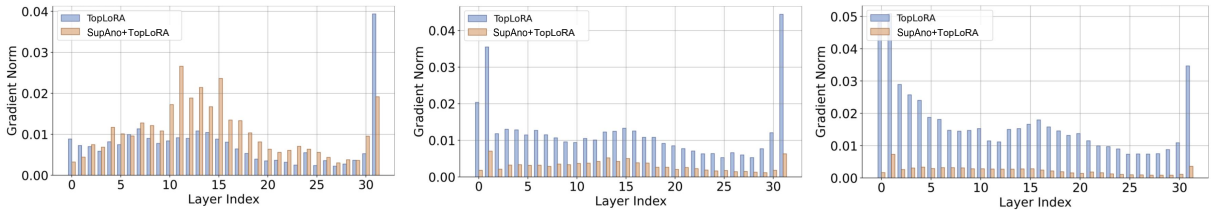

Figure 9: Comparison of gradient norms in different layers at training steps 3,200, 25,600, and 51,200 for TopLoRA and SupAno+TopLoRA.

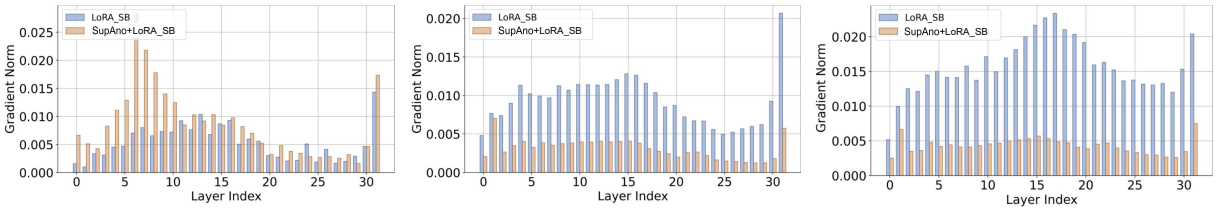

Figure 10: Comparison of gradient norms in different layers at training steps 3,200, 25,600, and 51,200 for LoRA-SB and SupAno+LoRA-SB.

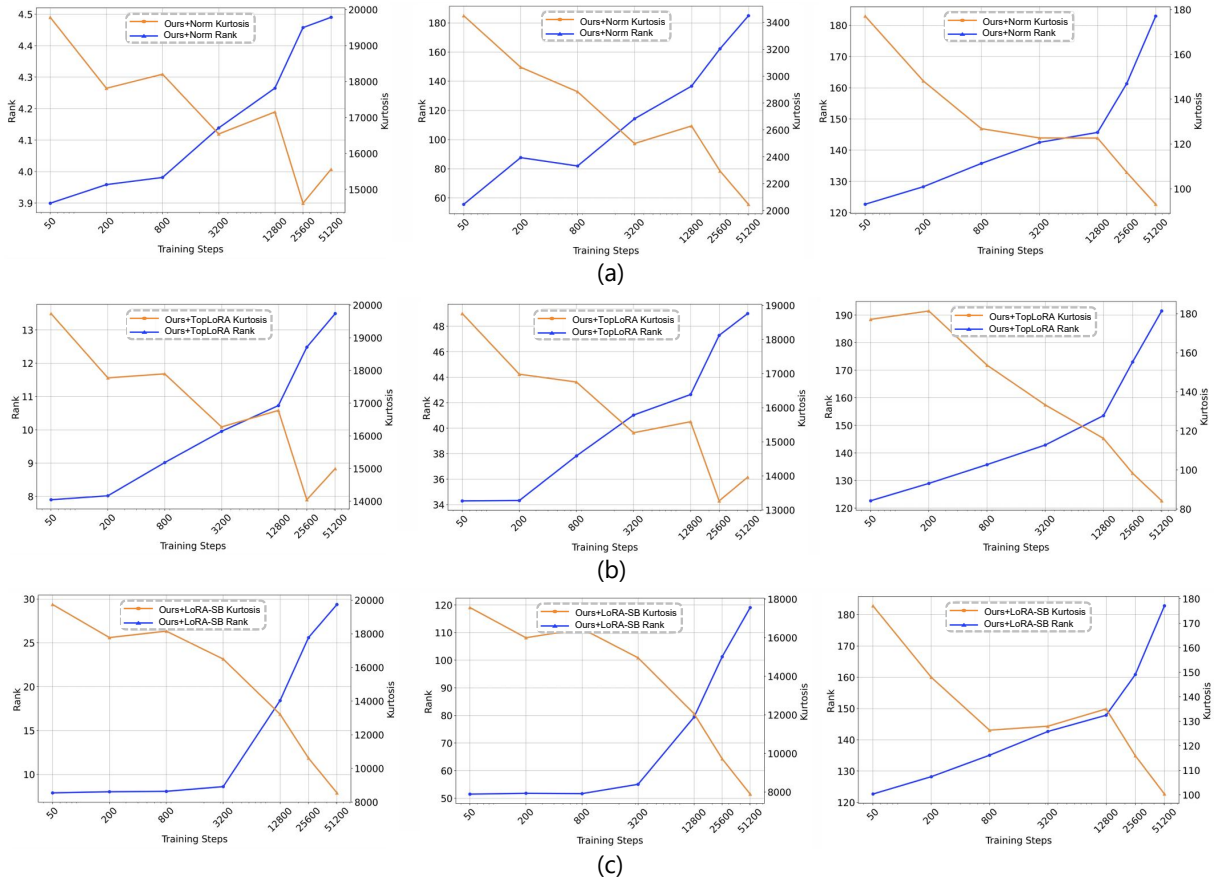

Figure 11: Evolution of activation kurtosis and effective rank in LLaMA2-7B layers 5, 14, and 32 during fine-tuning with SupAno+Norm (a), SupAno+TopLoRA (b), and SupAno+LoRA-SB (c).

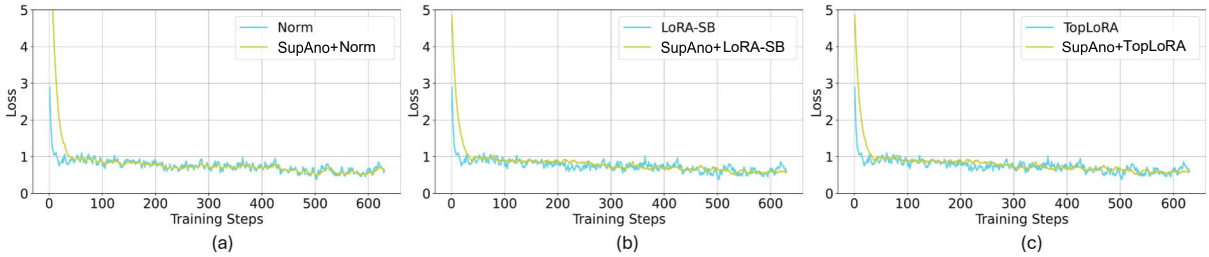

Figure 12: Training loss of Norm and SupAno+Norm (a), LoRA-SB and SupAno+LoRA-SB (b), and TopLoRA and SupAno+TopLoRA (c).