大规模模型低精度训练动力学理解与分析

姚权铭

清华大学 电子工程系

课题组介绍

核心主线：面向有限数据与算力约束，致力于解决智能体在资源受限场景下的高效学习与自主决策难题。我们构建从底层理论机理、自适应算法框架到工业与科学场景应用的全链条研究体系，探索数据高效范式下的通用人工智能新路径。

01 机理层 · 理论底座

研究大模型训练动力学，分析训练不稳定性成因，探索提升模型训练效率与稳定性的理论方法。

KEYWORDS：理论分析 / 数值稳定性

02 方法层 · 算法框架

聚焦智能体长程任务挑战，研究稳定演化算法，构建高效自适应的智能体学习框架与算法范式。

KEYWORDS：自适应优化 / 稳定演化

03 应用层 · 场景验证

面向国产算力栈部署通用智能体系统，拓展AI4S应用场景，赋能分子科学发现与药物研发。

KEYWORDS：国产化算力 / AI4S

代表性成果

1. Why Low-Precision Transformer Training Fails: An Analysis on Flash Attention (ICLR 2026)
2. Generalizing from a Few Examples: A Survey on Few-Shot Learning (ACM CSUR 2020)
3. Emerging Drug Interaction Prediction Enabled by Flow-based Graph Neural Network (Nature Computational Science 2023)
4. Co-teaching: Robust training deep neural networks with extremely noisy labels (NeurIPS 2018)
5. Why In-Context Learning Models are Good Few-Shot Learners? (ICLR 2025)

Data-Efficient Agentic Learning
牵头国际工作组，推动数据高效智能体学习领域发展。

https://cis.taskforce.ieee.org/deal/

报告大纲

Part I：低精度训练
浮点精度和大模型训练介绍

Part II：预测
谱对齐指标Spectral Alignment

Part III：归因
两个条件耦合导致爆炸

Part IV：修正
动态最大值的 SFA

Why Low-Precision Transformer Training Fails: An Analysis on Flash Attention. ICLR 2026 (oral)

Spectral Alignment as Predictor of Loss Explosion in Neural Network Training. Arvix 2025

大规模模型低精度训练动力学理解与分析

Understanding Large-Scale Model Training Dynamics
under Low Precision

Part I：低精度训练
浮点精度和大模型训练介绍

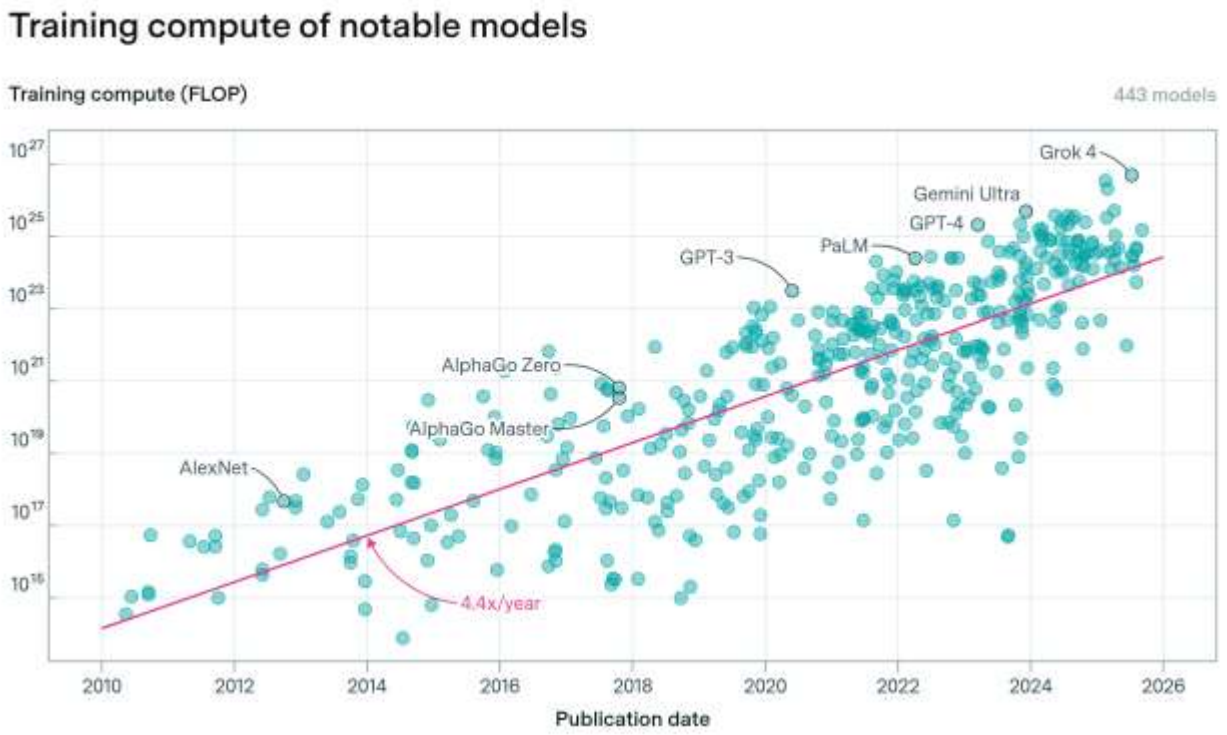
为什么要低精度训练

计算开销

大模型通常需要在海量数据上进行训练。随着模型参数量和训练数据规模不断增长，训练所需的计算量迅速增加。

显存开销

随着模型规模扩大，模型参数、梯度、优化器状态和中间激活值占用的显存持续增长，对硬件容量提出了更高要求。



Training compute for notable AI models has approximately doubled every six months since 2010.

而低精度训练通过降低数值表示的位宽，可以实现大模型训练的降本增效

为什么要低精度训练

低精度训练通过降低数值表示的位宽，减少计算量和显存占用，从而提高大模型训练效率、降低训练成本。

优势：

- 节省显存：BF16 张量的存储空间约为 FP32 的一半，可支持更大的 batch size、更长的序列或更大的模型。
- 加速计算：在支持低精度 Tensor Core 的硬件上，BF16 的理论峰值算力最高可达 FP32/TF32 的约 2 倍。

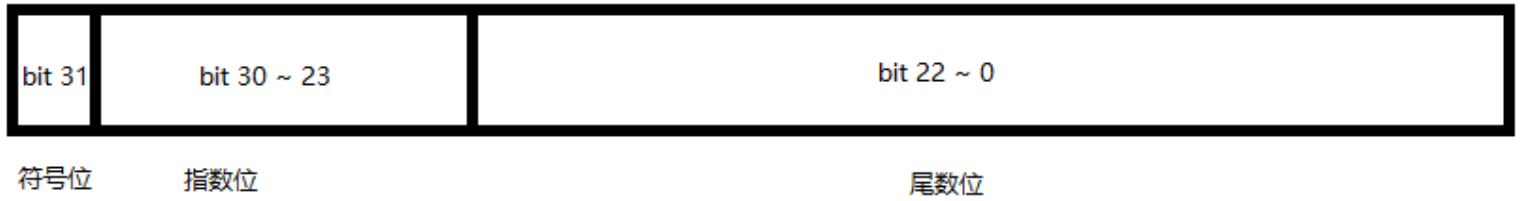
数值格式	理论算力(H100)	相对吞吐量
FP32	989 TFLOPS	1×
FP16/BF16	1979 TFLOPS	约 2×
FP8	3958 TFLOPS	约 4×

低精度的数值误差

计算机表示浮点数的一般格式为：

- s为符号位，f为尾数，e为指数，能表示的数为

$$x = (-1)^s \cdot (1 + f) \cdot 2^{e-bias}$$

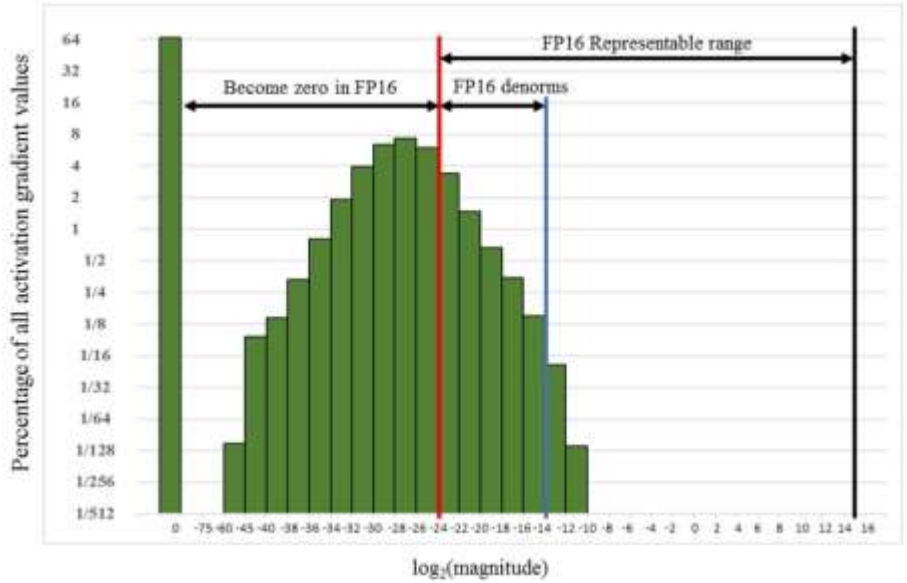


	总位数	指数位	尾数位	bias	有效精度	表示范围
FP16	16	5	10	15	约3-4位十进制	$[5.96 \times 10^{-8}, 6.55 \times 10^{4}]$ 带正负
FP32	32	8	23	127	约7位十进制	$[1.4 \times 10^{-45}, 3.4 \times 10^{38}]$ 带正负
BF16	16	8	7	127	约2-3位十进制	$[1.4 \times 10^{-45}, 3.4 \times 10^{38}]$ 带正负

- FP32太慢且消耗显存，FP16的范围太小，容易爆炸或者归零。BF16 在2021年由Google推广，其表示范围与FP32一致，精度比FP16更低。如今BF16已经成为LLM训练的主流格式

低精度的数值误差

上溢/下溢



运算舍入误差（比如大数吃小数）

两个数量级相差很大的数进行加减时，小数的有效信息落在大数浮点间隔以内，运算结果舍入后，小数完全或部分消失。

```
>>> 1e10+1e-10
10000000000.0
```

挑战：

- 数值误差可能在前向传播、反向传播和参数更新中不断累积或放大，造成训练不稳定！

后果：

对于千卡级训练，一次训练失败，可能导致回退到数天之前的检查点，造成：数万~十万小时的GPU-hours浪费；百万~千万美元金钱浪费；吉瓦时电力浪费；对团队士气的打击；对产品发布的延误

表示误差

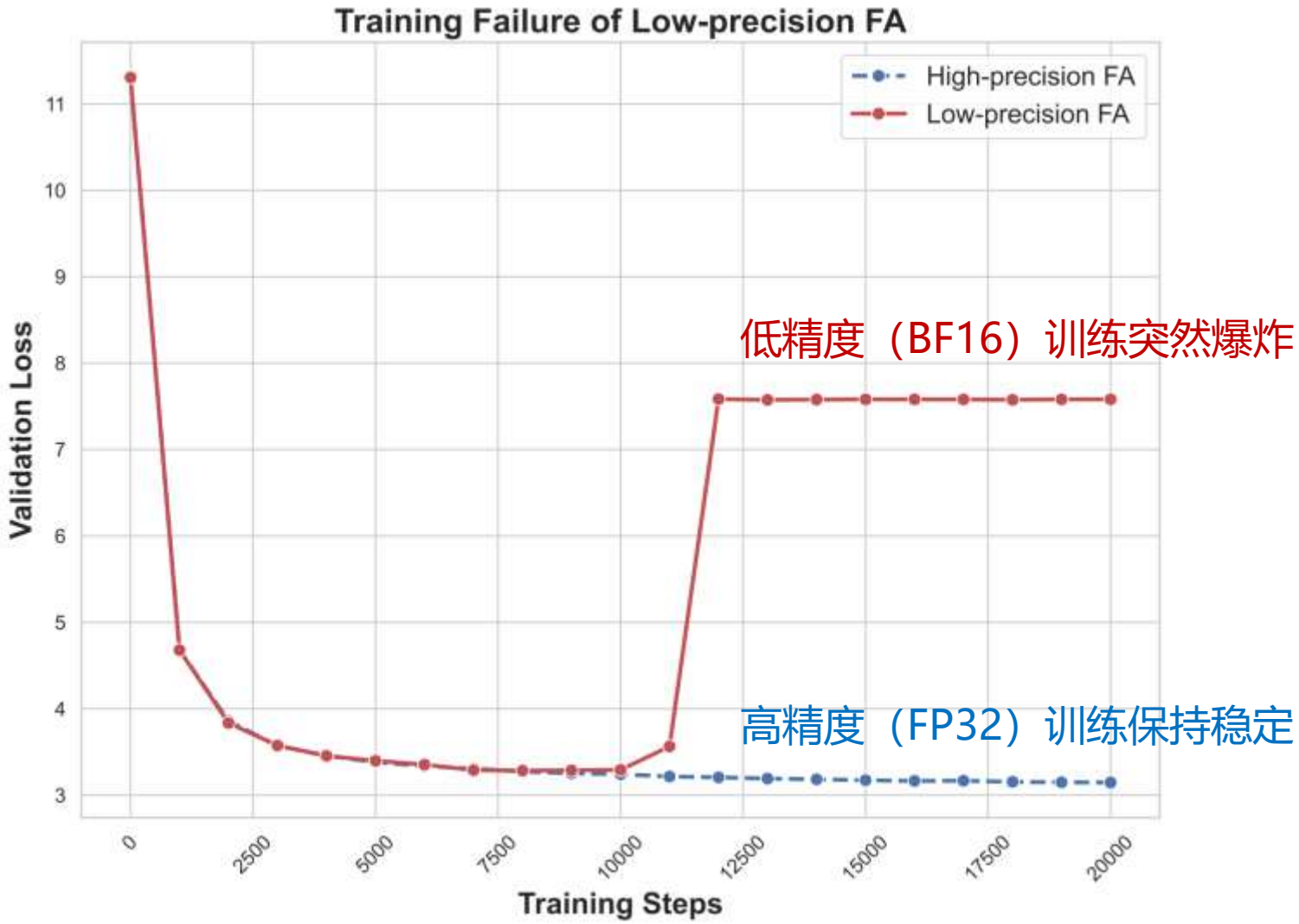
浮点数由有限个二进制位表示，可表示的数是离散的，因此多数实数只能舍入为邻近值。例如，0.1和0.2都无法用有限位二进制精确表示，导致：

$$\mathrm{fl}(0.1) = 0.1000000000000000555\cdots$$
$$\mathrm{fl}(0.2) = 0.2000000000000001110\cdots$$
$$\mathrm{fl}(\mathrm{fl}(0.1) + \mathrm{fl}(0.2)) = 0.3000000000000004$$

```
>>> 0.1+0.2
0.30000000000000004
```

失败案例：低精度 Flash Attention 让训练突然爆炸

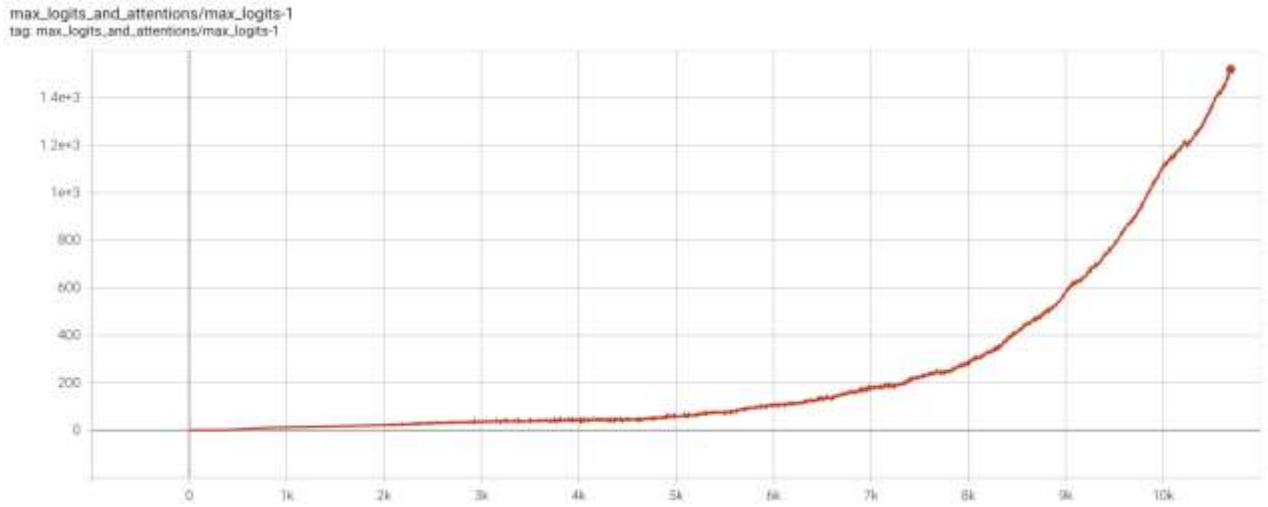
- 自2023年起，低精度FA导致训练爆炸被社区反复报告、但长期缺乏机制解释。（Gemma 2，Qwen 3 Next，Kimi K2中都有相似问题）
- 例如在大模型入门领域最具影响力的开源项目之一，nanoGPT的官方仓库中，大量Issue报告这一问题。（nanoGPT是由前特斯拉AI总监Karpathy打造、GitHub收获 61.7k Star 的 GPT-2 复现项目）
- 恢复高精度后训练稳定，说明原因是低精度数值误差导致的。



已有解决方案

$$X \to Q, K \to S = \frac{QK^\top}{\sqrt{d_k}} \to \bar{P} = \exp(S - m) \to O = \bar{P}V / \ell$$

方法	提出团队/模型	干预位置	核心作用
QK Norm	Henry et al.; Alibaba Qwen	Q, K	限制 Query、Key 的尺度
Logit Soft-capping	Google DeepMind · Gemma 2	S	限制 Softmax 前 logits
Gated Attention	Alibaba Qwen · Qwen3.5	O	限制 Attention 输出的信息流
QK-clip	Moonshot AI · Kimi K2	W_Q, W_K	从参数源头限制未来的 logits



尽管低精度 Transformer 的训练崩溃已被报告多年，现有稳定化方法大多仍是针对 QK、Attention logits 或输出幅值的经验性干预，却一直缺少一条能 **“从 Flash Attention 中的低精度数值误差出发,一路解释到 loss 爆炸”** 的完整机制链。

为什么分析低精度训练崩溃很难

我们面对的不是单一bug，而是一条从底层浮点误差到宏观loss爆炸的复杂机制链

链条太长

从一次 BF16 舍入，到最终 loss 爆炸，中间跨越多层计算与参数更新。

算子强耦合

Attention、Softmax、矩阵乘法、优化器彼此耦合，很难单独归因。

异常出现得晚

loss 往往在数千步后才突然崩溃，真正的异常早已在内部累积。

误差既小又隐蔽

单次数值误差几乎看不见，但可能在相似方向上持续累积并放大。

核心难点：打通“有偏 BF16 舍入误差 → 相似低秩梯度误差持续累积 → 权重谱范数与激活异常增长 → loss 爆炸”的完整机制链。

大规模模型低精度训练动力学理解与分析

Understanding Large-Scale Model Training Dynamics
under Low Precision

Part II：预测
谱对齐指标Spectral Alignment

如何预测训练崩溃

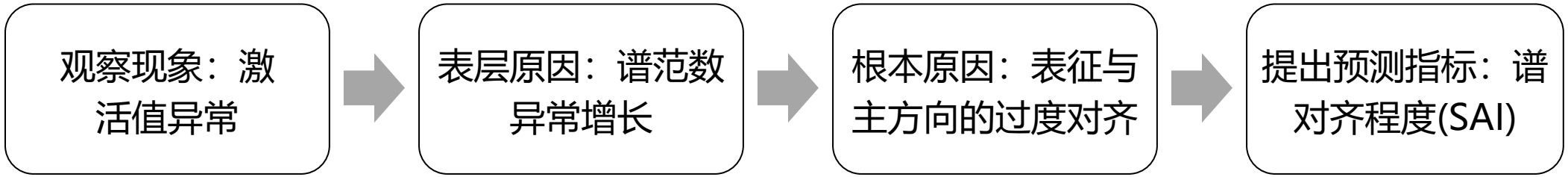
鉴于低精度训练可能导致的灾难性失败，开发能在低精度训练时预测不稳定发生的指标，已经成为深度学习领域巨大的需求和挑战。

已有方法：

- Fishman et al. 2025 (Intel)：SwiGLU中参数出现方向对齐，导致异常值放大
- Wortsman et al. 2024 (Deepmind)：softmax、LayerNorm和AdamW epsilon的相互作用，导致有效梯度消失
- Rybakov et al. 2024 (Nvidia)：线性层输出的范数持续增长，导致激活值异常

已有工作仅揭示了大谱范数、大激活、异常方向对齐与训练崩溃的相关性，并未定位到驱动训练走向崩溃的、更早期的本质原因

因果链分析思路：逐级解释谱范数增长到崩溃的逻辑链



低精度训练下，梯度数值下溢问题严重，参数矩阵被不均衡更新，导致输入表征被过度拉伸，加剧表征的病态对齐，使谱范数异常增长，最终导致训练崩溃。

如何预测训练崩溃：谱对齐驱动谱范数增长

理论分析结果：过度对齐导致崩溃

我们将一个表征与主方向的过度对齐状态，在数学上形式化为以下谱对齐条件：

条件 权重矩阵的主奇异向量 $\mathbf{u}_1$ 和 $\mathbf{v}_1$，与输入向量 $\mathbf{h}_{l-1}$ 和输出向量 $\mathbf{f}^{(l)}$ 之间存在高度共线性：

$$\mathbf{v}_1^\top = \alpha_1 \mathbf{f}^{(l)} + \boldsymbol{\epsilon}_1, \qquad \text{其中 } \alpha_1 = \left\langle \mathbf{f}^{(l)}, \mathbf{v}_1^\top \right\rangle, \qquad \left\langle \boldsymbol{\epsilon}_1, \mathbf{f}^{(l)} \right\rangle = 0$$

$$\mathbf{u}_1^\top = \alpha_2 \mathbf{h}^{(l-1)} + \boldsymbol{\epsilon}_2, \qquad \text{其中 } \alpha_2 = \left\langle \mathbf{h}^{(l-1)}, \mathbf{u}_1^\top \right\rangle, \qquad \left\langle \boldsymbol{\epsilon}_2, \mathbf{h}^{(l-1)} \right\rangle = 0$$

则有 $\|\boldsymbol{\epsilon}_1\| \ll \left\|\alpha_1 \mathbf{f}^{(l)}\right\|$ 成立，此时奇异向量几乎完全由其投影项来表示。

基于上述条件，对梯度下降更新过程中，权重谱范数变化进行了严格的数学推导，定量求出了训练过程中谱范数的增长变化量：

定理 权重矩阵 $\mathbf{W}_l$ 的谱范数 $\|\mathbf{W}_l\|_2$ 在单次梯度下降步骤中的增长 $\Delta\|\mathbf{W}_l\|_2$ 可以近似为：

$$\Delta\|\mathbf{W}_l\|_2 \approx -\eta\rho\alpha_1\alpha_2 \left\|\mathbf{h}^{(l-1)}\right\|_2^2 (\mathbf{p}-\mathbf{t})\left(\mathbf{h}^{(L)}\right)^\top$$

在由交叉熵损失 $\mathcal{L}$ 驱动的训练过程中，谱范数变化变化 $\Delta\|\mathbf{W}_l\|_2$ 为正，这意味着在对齐性条件下，谱范数会发生确定性的正向增长。

证明了**谱对齐是驱动谱范数增长，并导致训练崩溃的根本原因，观测对齐程度能够提供训练不稳定的早期预警信号**

如何预测训练崩溃：谱对齐的指标 Spectral Alignment

Spectral Alignment 的计算

选取待监控的线性层，其计算形式为：

$$y = Wx$$

其中 $x \in \mathbb{R}^{d_{\text{in}}}$是层输入表征，$W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ 是权重矩阵。

对权重矩阵 W进行奇异值分解：

$$W = U\Sigma V^{\top}$$

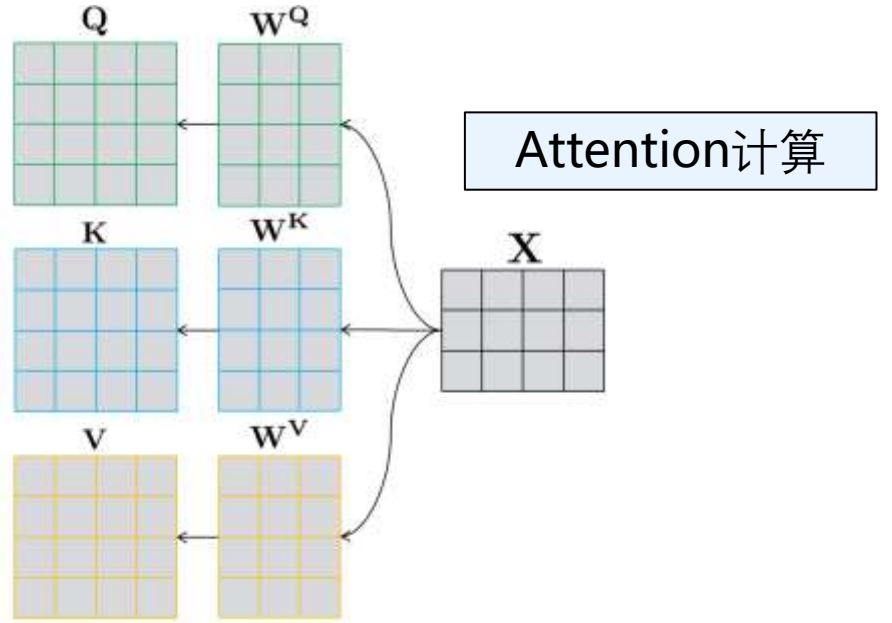
提取最大奇异值对应的**输入侧主奇异向量**：

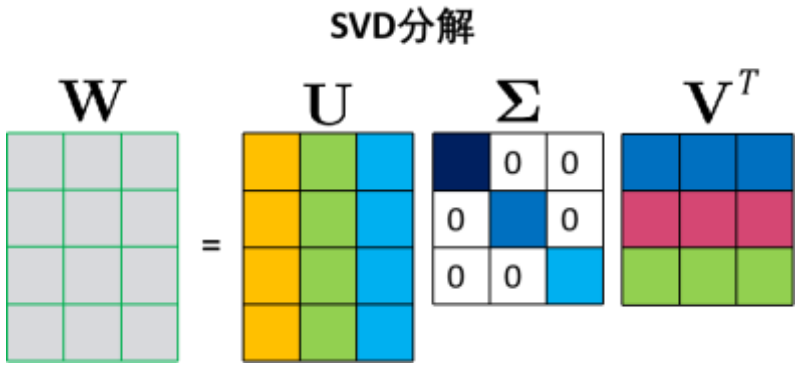
$$v_{max} = V[:, 1]$$

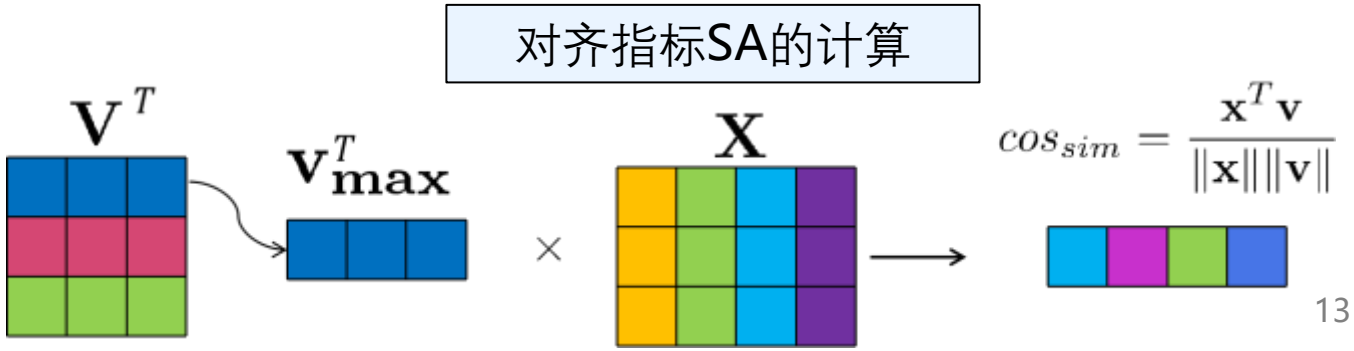
等价地，$v_{max}^{\top}$ 是 $V^{\top}$的第一行。

对每个输入表征 x_i，计算其与主奇异方向 v_{max} 的余弦相似度：

$$SA_i = \frac{x_i^{\top} v_{max}}{\| x_i \|_2 \| v_{max} \|_2}$$







预测效果

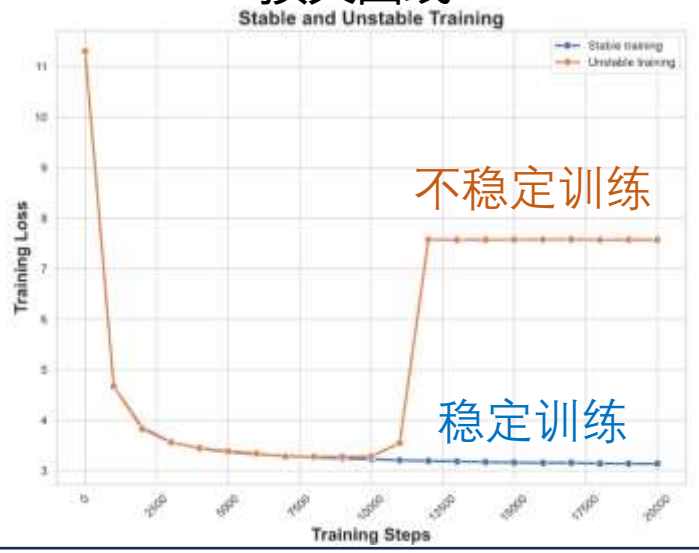
案例一：注意力层崩溃预警

案例：nanoGPT 模型在低精度 Flash Attention 训练中的崩溃

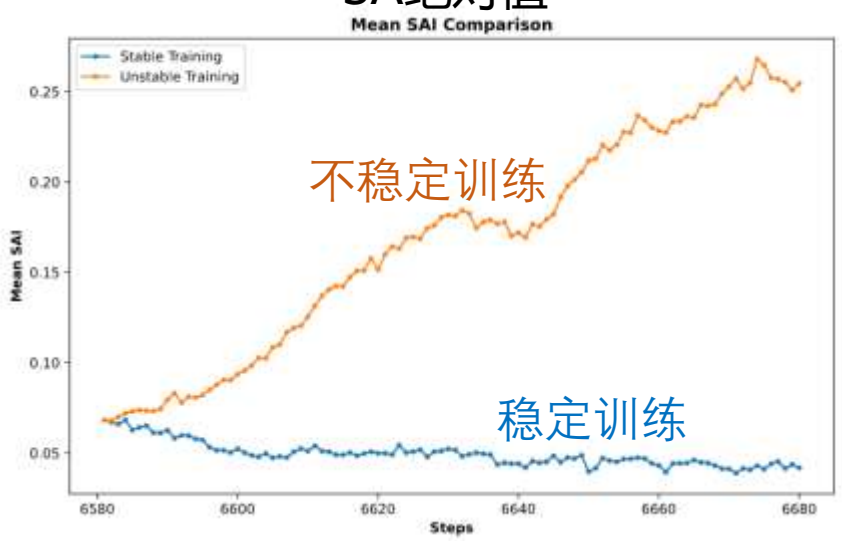
配置：nanoGPT 模型/ bf16 精度

SAI 指标：在损失爆炸 4000 步前发生整体分布的偏移，绝对平均值异常上升，发出明确的早期预警

损失曲线



SA绝对值

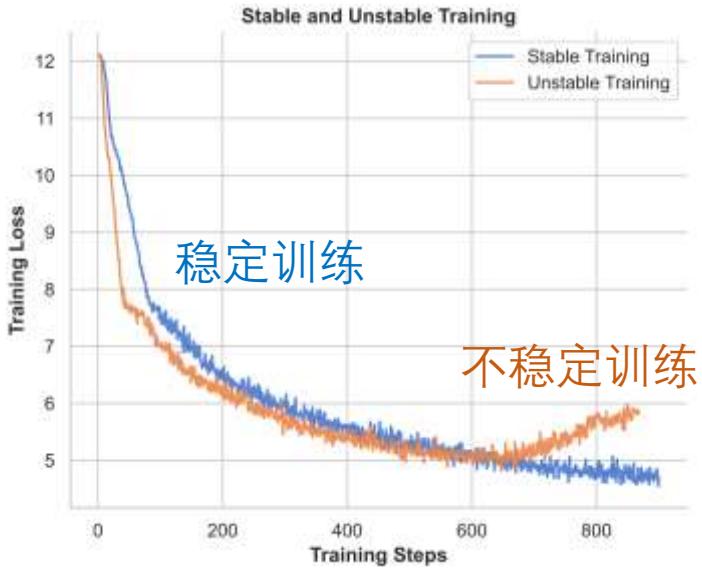


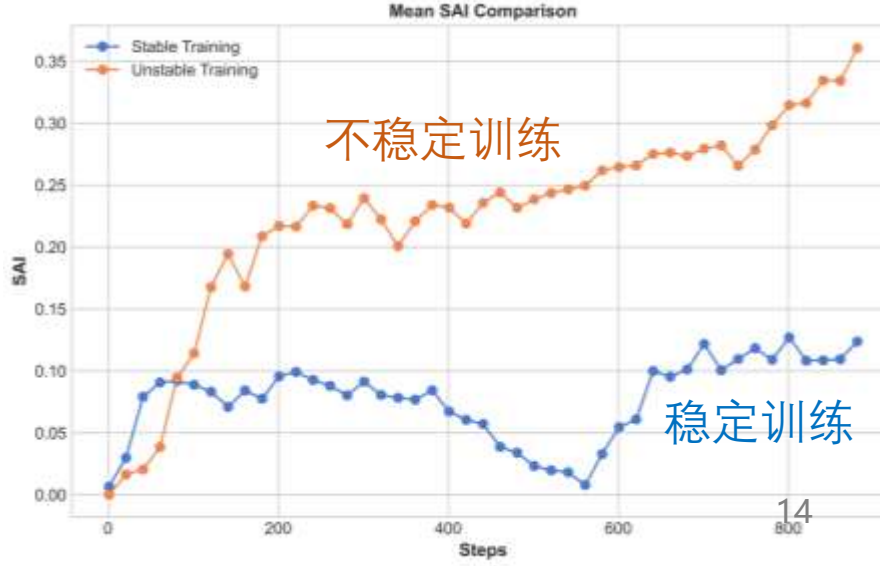
在低精度FA之外的多种训练不稳定场景下均展现有效性

案例二：前馈层崩溃预警

案例：Qwen2.5 – 0.5B 模型 FFN 层在大学习率下的训练崩溃

SAI 指标：在损失爆炸 300 步前发生整体分布的偏移，绝对平均值提前 500 步异常上升



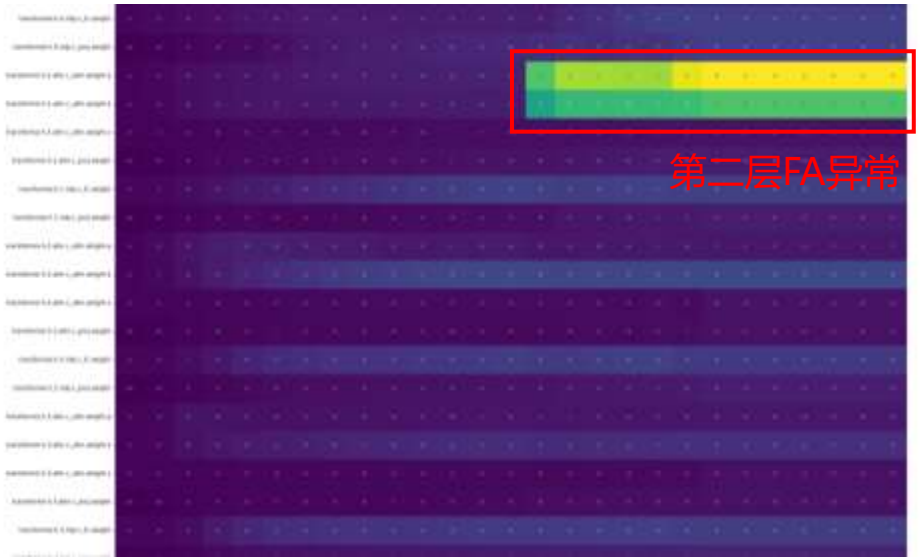


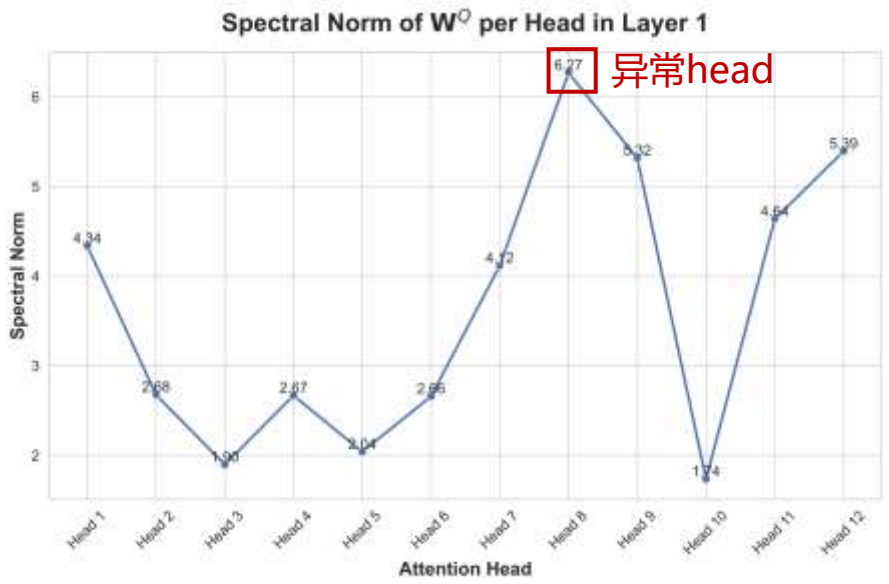
大规模模型低精度训练动力学理解与分析

Understanding Large-Scale Model Training Dynamics
under Low Precision

Part Ⅲ：归因
两个条件耦合导致爆炸

误差入口：低精度算子$\boldsymbol{\delta} = \mathrm{rowsum}(\mathbf{d}O \circ \mathbf{O})$





层定位：第二层 FA 足以导致崩溃

头定位：有较大谱范数的head导致崩溃

算子级定位

- Flash Attention 反向传播涉及一个关键中间量$\boldsymbol{\delta}$

$$\boldsymbol{\delta} = \mathrm{rowsum}(\mathbf{d}O \circ \mathbf{O}).$$

将其恢复高精度后训练稳定，说明低精度数值误差入口为算子$\boldsymbol{\delta}$

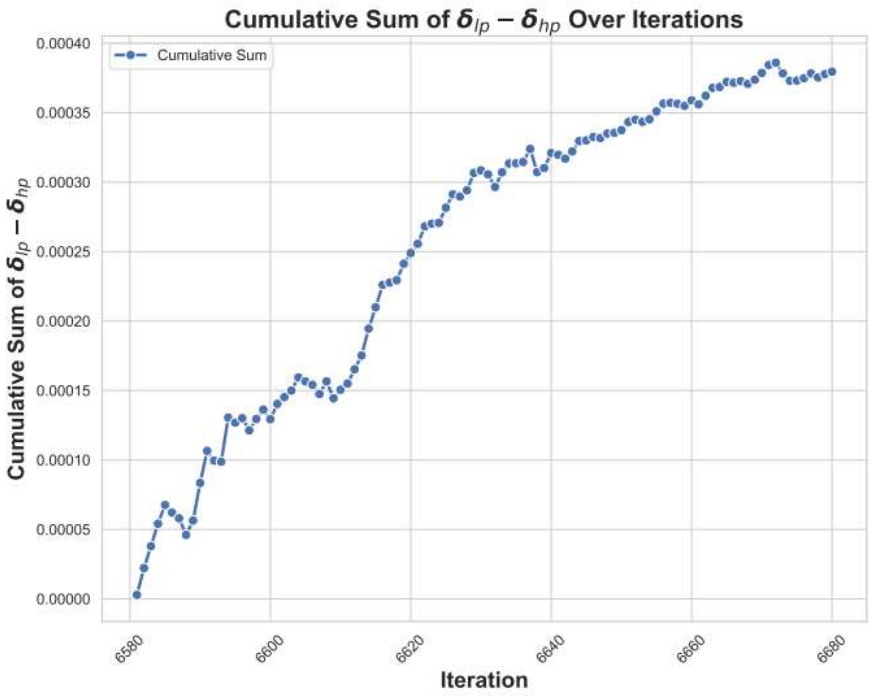
$\mathbf{W}^Q$更新失衡：两种偏置相互耦合

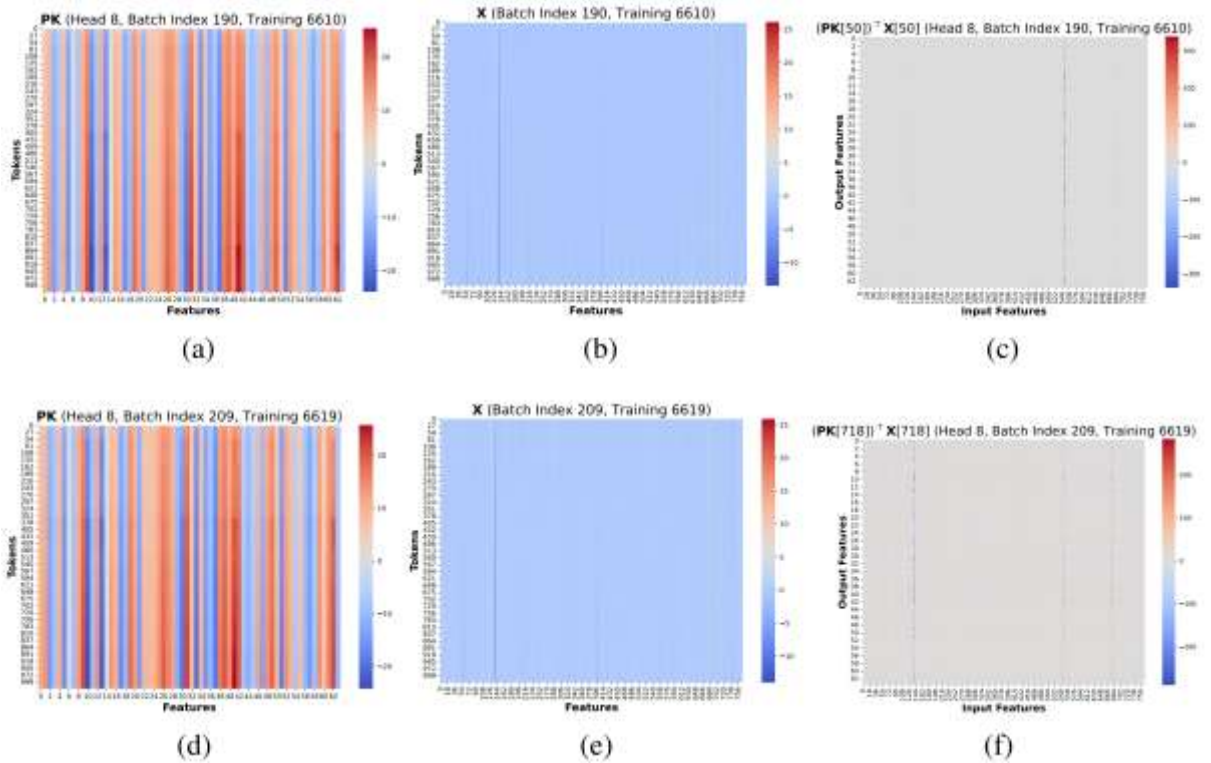
- $\mathbf{W}^Q$在高低精度下数值差异表达式中，包含两种系统性偏置，其相互耦合导致$\mathbf{W}^Q$更新失衡

$$\mathbf{d}W_{\mathrm{hp}}^{Q} - \mathbf{d}W_{\mathrm{lp}}^{Q} = \alpha \sum_{T=1}^{N} (\boldsymbol{\delta}_{\mathrm{lp}} - \boldsymbol{\delta}_{\mathrm{hp}})[T] (\mathbf{PK})[T]^{\top} \mathbf{X}[T]$$

BF16舍入误差导致的偏置

相似低秩结构导致的偏置





$\sum_{T=1}^{N}(\boldsymbol{\delta}_{\mathrm{lp}} - \boldsymbol{\delta}_{\mathrm{hp}})[T]$出现明显的正向偏置

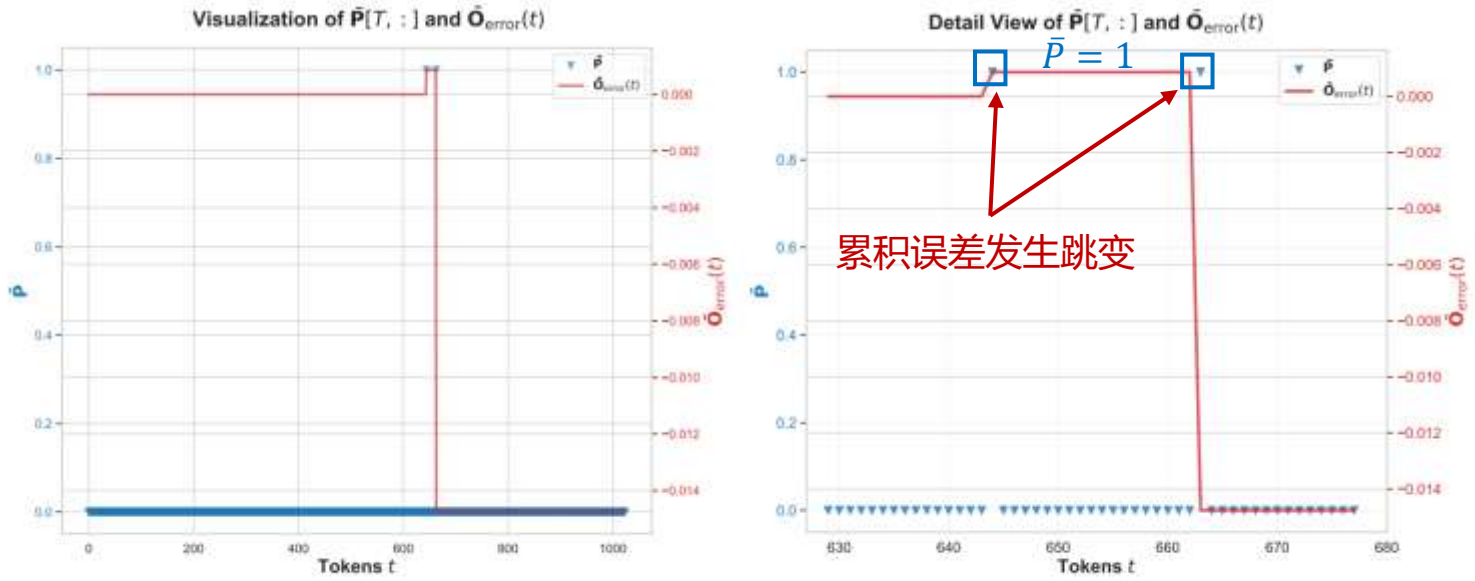
$(\mathbf{PK})[T]^{\top}\mathbf{X}[T]$ 的低秩结构相似性

$\sum_{T=1}^{N}(\boldsymbol{\delta}_{\mathrm{lp}}-\boldsymbol{\delta}_{\mathrm{hp}})[T]$偏置：$\bar{\mathbf{O}}=\bar{\mathbf{P}}\mathbf{V}$低精度运算

- 进行消融分析，发现$\sum_{T=1}^{N}(\boldsymbol{\delta}_{\mathrm{lp}}-\boldsymbol{\delta}_{\mathrm{hp}})[T]$ 的偏置来源于FA计算softmax时使用的$\bar{\mathbf{O}}=\bar{\mathbf{P}}\mathbf{V}$运算

$$\bar{\mathbf{P}}=\exp\big(\mathbf{S}-\mathrm{rowmax}(\mathbf{S})\big),\ \bar{\mathbf{O}}=\bar{\mathbf{P}}\mathbf{V},\ \mathbf{O}=\frac{\bar{\mathbf{O}}}{\mathrm{rowsum}(\bar{\mathbf{P}})}.$$

- 进一步实验发现，累积和$\sum_{T=1}^{N}(\boldsymbol{\delta}_{\mathrm{lp}}-\boldsymbol{\delta}_{\mathrm{hp}})[T]$发生跳变时刻，恰好对应$\bar{P}=1$时刻



(b) Large error when $\bar{\mathbf{P}}[T,i]=1$　　(c) Detailed view of (b)

累积误差$O_{\mathrm{lp}}[T,i]-O_{\mathrm{hp}}[T,i]$发生跳变的时刻（红色线波动），恰好对应$\bar{P}=1$（蓝色三角位于顶端）

$\overline{\mathbf{P}} = 1$时刻产生较大向上舍入误差

- $\overline{\mathbf{P}} = 1$时，对应的$\overline{\mathbf{P}}$V值为低精度，在FP32高精度表示中，后16位均为0

1 10000000 00110100000111000101110 (-2.4071154594421387)
1 10000000 00100110000000000000000 (-2.296875)

符号位 指数位 BF16尾数位 FP32尾数位 精确十进制数值

$\overline{\mathbf{P}} = 1$ 时刻之前的累积和（FP32尾数位非零）
$\overline{\mathbf{P}} = 1$ 时刻对应的V值（FP32尾数位为零）

- 此时用FP32高精度计算累积和，再舍入到BF16低精度

1 10000001 00101101000011100010111 (-4.703990459442139)
1 10000001 0010111 (-4.71875)

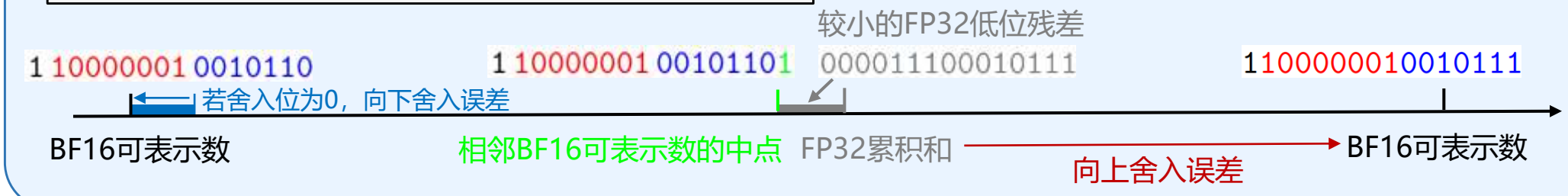
FP32累积和（舍入位为1，BF16将向上舍入）
BF16舍入（BF16尾数位最后一位从0变成1）

- 向上舍入带来了较高的舍入精度误差

假如此时舍入位为0，向下舍入误差仅为0.00087，远小于向上舍入

1 10000001 0010111 - 1 10000001 00101101000011100010111 = -0.014759540557861328

较小的FP32低位残差触发较大的向上舍入误差



不平衡的舍入误差

- 较大向上舍入误差产生的条件
  - I. 至少出现两次 $\overline{\mathrm{P}}=1$，且其它的 $\overline{\mathrm{P}}$ 值很小
  - II. 两次 $\overline{\mathrm{P}}=1$ 对应的V值数值接近

第一次 $\overline{\mathbf{P}}=1$

其它的 $\overline{\mathrm{P}}$ 值很小，FP32尾数位中仅有低位残差

1100000000011010 0000111000101110

第二次 $\overline{\mathbf{P}}=1$，FP32尾数位均为0

1100000000010011 0000000000000000

相近的V值使指数位对齐

舍入位为 1

1 10000001 00101101 0000111000101110

向上舍入

1100000010010111

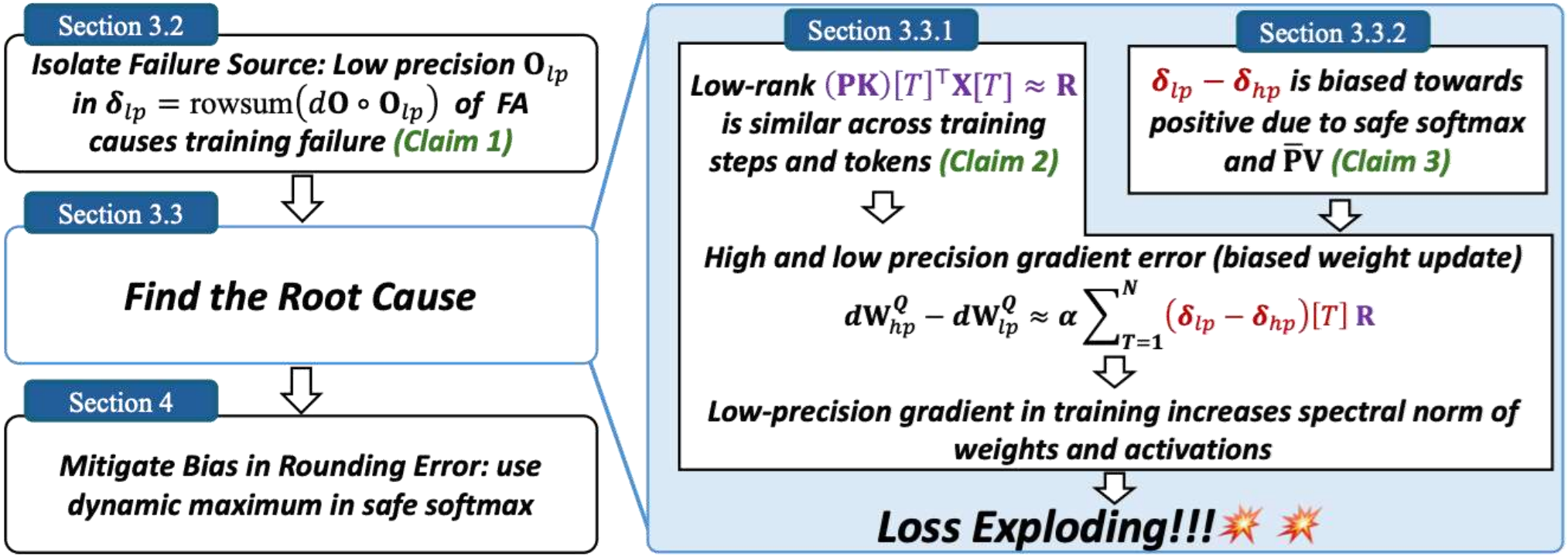
尽管舍入位为1或0的概率近似相同，
但是向上舍入数值远大于向下舍入，
导致$\sum_{T=1}^{N}(\boldsymbol{\delta}_{\mathrm{lp}}-\boldsymbol{\delta}_{\mathrm{hp}})[T]$出现明显正向偏置

根因总结：两个偏置相互耦合，最终导致Loss爆炸

BF16舍入误差导致的偏置　相似低秩更新方向导致的偏置

$$\mathbf{d}W_{\mathrm{hp}}^{Q} - \mathbf{d}W_{\mathrm{lp}}^{Q} = \alpha \sum_{T=1}^{N} (\boldsymbol{\delta}_{\mathrm{lp}} - \boldsymbol{\delta}_{\mathrm{hp}})[T]\,(\mathbf{PK})[T]^{\top}\mathbf{X}[T]$$

→ **低精度下梯度$\mathbf{d}W_{\mathrm{lp}}^{Q}$在同一方向累积** → **$\mathbf{W}^{Q}$ 谱范数异常增大，最终导致Loss爆炸**



Loss 爆炸的逻辑链条

大规模模型低精度训练动力学理解与分析

Understanding Large-Scale Model Training Dynamics
under Low Precision

Part IV：修正
动态最大值的 SFA

修正算法：SFA，Stabilized Flash Attention

- Flash Attention 中标准safe softmax计算为

$$\mathbf{r}_m = \mathrm{rowmax}(\mathbf{S}), \bar{\mathbf{P}} = \exp(\mathbf{S} - \mathbf{r}_m), \bar{\mathbf{O}} = \bar{\mathbf{P}}\mathbf{V}, \mathbf{O} = \frac{\bar{\mathbf{O}}}{\mathrm{rowsum}(\bar{\mathbf{P}})}.$$

  如果一行中最大值$\mathbf{r}_m$出现至少两次，则其均对应有$\bar{\mathbf{P}} = 1$，从而引起不平衡舍入

- 修正思路：增大$\bar{\mathbf{P}} = \exp(\mathbf{S} - \mathbf{r}_m)$中的平移项$\mathbf{r}_m$，使得$\mathbf{S} - \mathbf{r}_m$计算结果小于0，导致$\bar{\mathbf{P}} < 1$，从而破坏$\bar{\mathbf{P}} = 1$这一触发条件
- 提出SFA（Stabilized Flash Attention，稳定快速注意力）算法，动态调整softmax计算中的行最大值$\mathbf{r}_m$，使得后16位中高位的0,1分布均匀，实现低精度稳定训练

$$\mathbf{r}_m = \mathrm{rowmax}(\mathbf{S}), \qquad \mathbf{r}_s = \mathrm{rowsum}(\mathbf{r}_m - \mathbf{S} \le \epsilon),$$

$$\mathbf{m}_{\mathrm{dyn}} = \begin{cases} \beta\mathbf{r}_m, & \mathbf{r}_s > 1 \wedge \mathbf{r}_m > 0, \\ 0, & \mathbf{r}_s > 1 \wedge \mathbf{r}_m < 0, \\ \mathbf{r}_m, & \text{otherwise}, \end{cases}$$

$$\bar{\mathbf{P}} = \exp(\mathbf{S} - \mathbf{m}_{\mathrm{dyn}}).$$

Practical setting: $\beta = 2, \epsilon \approx 10^{-3}$.

完整SFA算法

Algorithm 3 Stabilized Flash Attention by Mitigating Biased Rounding Error: Forward Pass

Require: Matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d}$, block sizes B_c, B_r, $\beta > 1$, $\epsilon > 0$.

Divide $\mathbf{Q}$ into $T_r = \left\lceil \frac{N}{B_r} \right\rceil$ blocks $\mathbf{Q}_1, \ldots, \mathbf{Q}_{T_r}$ of size $B_r \times d$ each, and divide $\mathbf{K}, \mathbf{V}$ in to $T_c = \left\lceil \frac{N}{B_c} \right\rceil$ blocks $\mathbf{K}_1, \ldots, \mathbf{K}_{T_c}$ and $\mathbf{V}_1, \ldots, \mathbf{V}_{T_c}$, of size $B_c \times d$ each.

Divide the output $\mathbf{O} \in \mathbb{R}^{N \times d}$ into T_r blocks $\mathbf{O}_1, \ldots, \mathbf{O}_{T_r}$ of size $B_r \times d$ each, and divide the logsumexp $\mathbf{L}$ into T_r blocks $\mathbf{L}_1, \ldots, \mathbf{L}_{T_r}$ of size B_r each.

for $1 \le i \le T_r$ **do**

  Initialize $\mathbf{O}_i^{(0)} = (0)_{B_r \times d} \in \mathbb{R}^{B_r \times d}, \boldsymbol{\ell}_i^{(0)} = (0)_{B_r} \in \mathbb{R}^{B_r}, \mathbf{m}_i^{(0)} = (-\infty)_{B_r} \in \mathbb{R}^{B_r}$.

  for $1 \le j \le T_c$ **do**

    Compute $\mathbf{S}_i^{(j)} = \mathbf{Q}_i \mathbf{K}_j^T \in \mathbb{R}^{B_r \times B_c}$.

    $\mathbf{r}_m = \mathrm{rowmax}(\mathbf{S}_i^{(j)}), \mathbf{r}_s = \mathrm{rowsum}(\mathbf{r}_m - \mathbf{S}_i^{(j)} \le \epsilon)$

    $\mathbf{m}_i^{(j)\prime} = \mathrm{where}(\mathbf{r}_m > 0 \wedge \mathbf{r}_s > 1, \beta \mathbf{r}_m, \mathbf{r}_m)$

    $\mathbf{m}_i^{(j)\prime} = \mathrm{where}(\mathbf{r}_m < 0 \wedge \mathbf{r}_s > 1, 0, \mathbf{m}_i^{(j)\prime})$

    Compute $\mathbf{m}_i^{(j)} = \max(\mathbf{m}_i^{(j-1)}, \mathbf{m}_i^{(j)\prime}) \in \mathbb{R}^{B_r}$, $\tilde{\mathbf{P}}_i^{(j)} = \exp(\mathbf{S}_i^{(j)} - \mathbf{m}_i^{(j)}) \in \mathbb{R}^{B_r \times B_c}$ (pointwise), $\boldsymbol{\ell}_i^{(j)} = e^{\mathbf{m}_i^{(j-1)} - \mathbf{m}_i^{(j)}} \boldsymbol{\ell}_i^{(j-1)} + \mathrm{rowsum}(\tilde{\mathbf{P}}_i^{(j)}) \in \mathbb{R}^{B_r}$.

    Compute $\mathbf{O}_i^{(j)} = \mathrm{diag}(e^{\mathbf{m}_i^{(j-1)} - \mathbf{m}_i^{(j)}}) \mathbf{O}_i^{(j-1)} + \tilde{\mathbf{P}}_i^{(j)} \mathbf{V}_j$.

  end for

  Compute $\mathbf{O}_i = \mathrm{diag}(\boldsymbol{\ell}_i^{(T_c)})^{-1} \mathbf{O}_i^{(T_c)}$.

  Compute $\mathbf{L}_i = \mathbf{m}_i^{(T_c)} + \log(\boldsymbol{\ell}_i^{(T_c)})$.

  Write $\mathbf{O}_i$ as the i-th block of $\mathbf{O}$.

  Write $\mathbf{L}_i$ as the i-th block of $\mathbf{L}$.

end for

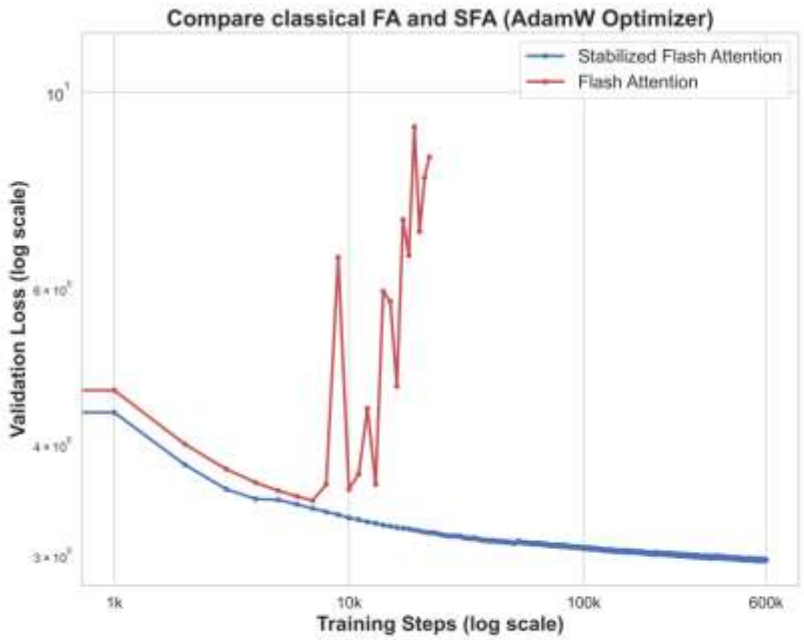
Return the output $\mathbf{O}$ and the logsumexp $\mathbf{L}$.

仅进行轻量级插入式修正

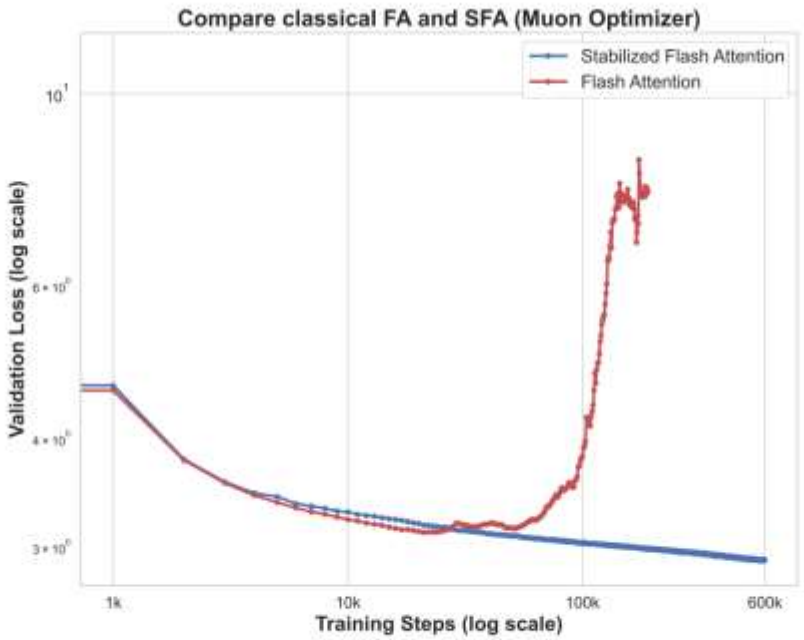
SFA算法，相比标准FA，仅在softmax计算前进行插入式修正，无需更改FA反向传播

实验效果：SFA 阻止loss爆炸

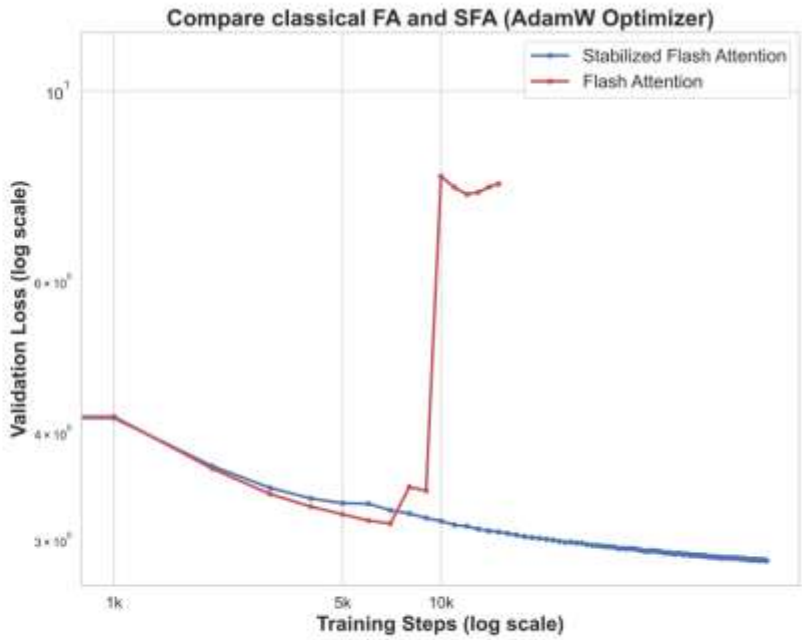
- 在不同规模的模型（GPT-2S、GPT-2M），和不同的优化器（AdamW、Muon）设置下，SFA均阻止了loss爆炸，实现稳定训练



(a) GPT-2S+AdamW



(b) GPT-2S+Muon



(c) GPT-2M+AdamW

标准 FA 与 SFA 的验证 loss 对比，SFA使得训练恢复稳定

总结

低精度训练稳定性的“预测—归因—修正”闭环：

何时会崩？ → 为什么会崩？ → 如何修正？

预测：何时会崩?
Spectral Alignment（SA)

- 监控输入表征与权重主奇异方向的对齐分布
- 持续的单侧坍缩早于 Loss Explosion 出现
- 为风险预警和健康检查点选择提供依据

归因：为什么会崩?
有偏舍入误差 × 相似低秩结构

- Flash Attention 的 BF16 $\bar{P}V$ 计算产生系统性舍入偏差
- 相似低秩结构使梯度误差沿固定方向累积，而非相互抵消
- 推动权重谱范数和激活异常增长，最终导致 Loss Explosion

修正：如何避免崩溃?
Stabilized Flash Attention(SFA)

- 在出现多个或近似最大 Attention Score 时，动态调整 Safe Softmax 的平移量
- 避免危险位置出现 $\bar{P} = 1$，削弱有偏舍入误差
- 保持精确算术下的数学等价性，并在实验中恢复稳定训练

影响力

StoSignSGD: Unbiased Structural Stochasticity Fixes SignSGD for Training Large Language Models

Dingzhi Yu* Rui Pan* Yuxing Liu* Tong Zhang
University of Illinois Urbana-Champaign
dingzhiyu01@gmail.com, {yuxing6, ruip4, tozhang}@illinois.edu
*Equal Contribution

At the same time, another line of work studies why low-precision training fails and how to stabilize it. Wortsman et al. [2023] identified loss spikes associated with under-estimated AdamW second moments in large-scale low-precision training. Lee et al. [2024] systematically quantified the degradation in LLM training stability caused by aggressive precision reduction, while Balança et al. [2024] formalized scale propagation as an end-to-end recipe for low-precision LLM training. Most recently, Qiu and Yao [2026] traced an important failure mode of low-precision Transformer training to FlashAttention [Dao et al., 2022, Dao, 2024, Shah et al., 2024]. In contrast to these works, our goal is not to design yet another stabilization recipe for AdamW, but to show that replacing AdamW with a sign-based optimizer can itself provide a simple and robust alternative in highly constrained low-precision regimes.

KIMI K3: OPEN FRONTIER INTELLIGENCE

TECHNICAL REPORT OF KIMI K3

Kimi Team

ABSTRACT

We introduce Kimi K3, a 2.8T parameter Mixture-of-Experts model with 104 billion activated parameters, native vision capabilities, and a 1-million-token context window. Kimi K3 is built on Kimi Delta Attention [63] and Attention Residuals [57], which improve information flow across

To correct the biased rounding error that arises in flash attention, we adopt the method of [98] and k output in FP32 during training. This choice doubles the on-chip footprint of the output tile, we there training kernel to overlap it with the KV staging buffers instead of the query tile, freeing shared mer KV pipeline and higher training throughput.

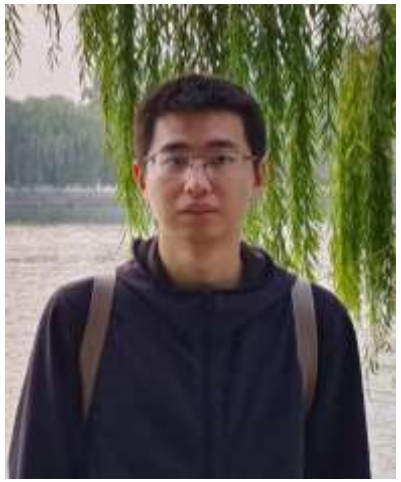
[97] Zhen Qin et al. *HGRN2: Gated Linear RNNs with State Expansion*. 2024. arXiv: 2404.07904 [cs.CL]. URL: https://arxiv.org/abs/2404.07904.

[98] Haiquan Qiu and Quanming Yao. "Why Low-Precision Transformer Training Fails: An Analysis on Flash Attention". In: *International Conference on Learning Representations (ICLR)*. 2026. arXiv: 2510.04212 [cs.LG].

[99] Zihan Qiu et al. *Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free*. 2025. arXiv: 2505.06708 [cs.CL].

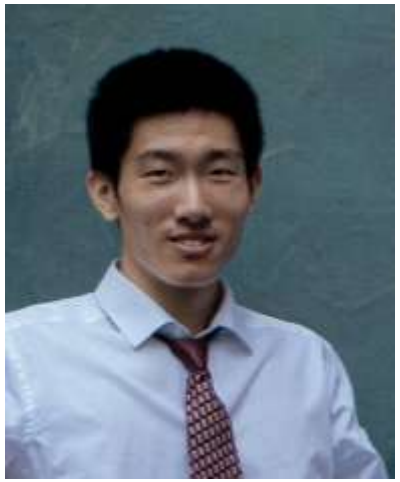
[100] Samyam Rajbhandari et al. "Zero: Memory optimizations toward training trillion parameter models". In: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2020,

后续工作不仅引用了本文结论，还直接采用本文机制模型设计训练预警指标，并复用本文代码与实验设置研究新的低精度优化方法。



邱海权

PhD student
qhq22@mails.tsinghua.edu.cn



吴尤

MS student
wu-you21@mails.tsinghua.edu.cn

谭英杰

PhD student
tanyj23@mails.tsinghua.edu.cn

谢谢!