Muon 可扩展用于大语言模型训练

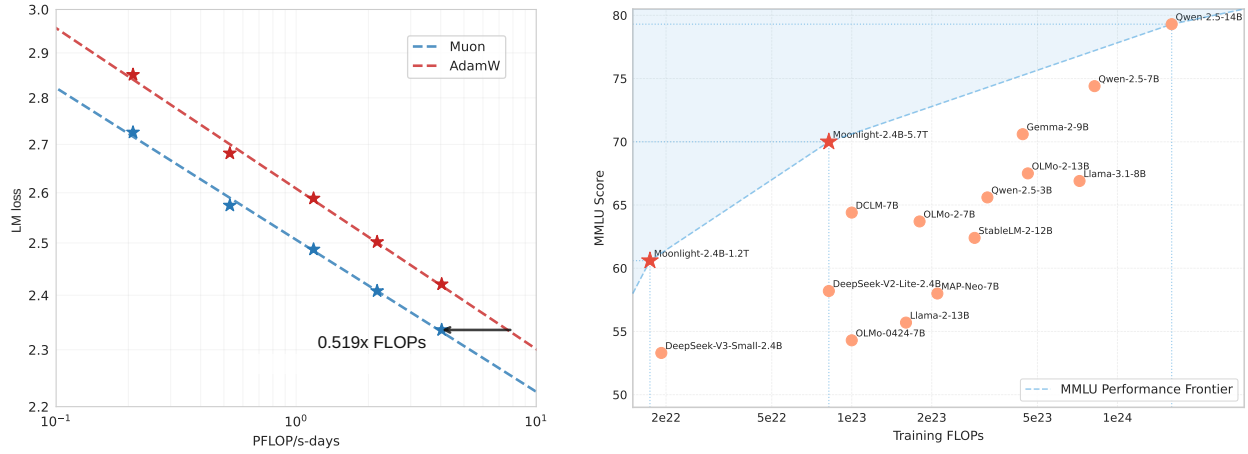
Technical Report

Jingyuan Liu[1] Jianlin Su[1] Xingcheng Yao[2] Zhejun Jiang[1] Guokun Lai[1] Yulun Du[1]
Yidao Qin[1] Weixin Xu[1] Enzhe Lu[1] Junjie Yan[1] Yanru Chen[1] Huabin Zheng[1]
Yibo Liu[1] Shaowei Liu[1] Bohong Yin[1] Weiran He[1] Han Zhu[1] Yuzhi Wang[1]
Jianzhou Wang[1] Mengnan Dong[1] Zheng Zhang[1] Yongsheng Kang[1] Hao Zhang[1]
Xinran Xu[1] Yutao Zhang[1] Yuxin Wu[1] Xinyu Zhou[1] * Zhilin Yang[1]

[1] Moonshot AI [2] UCLA

Abstract

最近，基于矩阵正交化的 Muon 优化器 (jordan2024muon) 在训练小规模语言模型方面展示了强劲的性能，但其在更大规模模型上的可扩展性尚未得到验证。我们确定了两个对 Muon 进行扩展的关键技术：(1) 添加权重衰减和 (2) 精心调整每个参数的更新比例。这些技术使得 Muon 能够无需调参即可在大规模训练中开箱即用。缩放法则实验表明，与 AdamW 相比，Muon 在计算最优训练下实现了 ∼2× 的计算效率。基于这些改进，我们推出了 Moonlight，这是一个使用 Muon 训练、包含 3B/16B 参数的 MoE 模型，使用了 5.7T token 进行训练。我们的模型提升了当前的帕累托前沿，与先前的模型相比，在显著减少训练 FLOPs 的同时实现了更好的性能。我们开源了内存优化且通信高效的分布式 Muon 实现。同时，我们发布了预训练、指令调优及中间检查点，以支持未来的研究。



(a) (b)

图 1: 使用 Muon 进行扩展。(a) 比较 Muon 和 Adam 的缩放法则实验。在计算最优训练的情况下，Muon 的计算效率比 Adam 高 ∼ 2× 。(b) 我们用 Muon 优化的 Moonlight 模型以及其他可比模型的 MMLU 性能。Moonlight 推进了性能与训练 FLOP 的帕累托前沿。

*Corresponding author: zhouxinyu@moonshot.cn

1 引言

大型语言模型（LLMs） (openai2024gpt4technicalreport; deepseekai2024deepseekv3technicalreport; grattafiori2024llama3herdmodels; geminiteam2024geminifamilyhighlycapable) 的快速发展极大地推动了通用人工智能的进步。然而，由于缩放法则 (kaplan2020scalinglawsneurallanguage; hoffmann2022trainingcomputeoptimallargelanguage)，训练强大的 LLMs 仍然是一个计算密集型和资源需求高的过程。优化器在高效且有效地训练 LLMs 中起着至关重要的作用，其中 Adam (adam2015kingma) 及其变体 AdamW (loshchilov2018decoupled) 是大多数大规模训练的标准选择。

近期最优化算法的发展显示出超越 AdamW (liu2024sophia; jordan2024muon; yuan2024mars; vyas2025soap; Li_2018; li2018preconditionermatrixliegroup; pooladzandi2024curvatureinformedsgdgeneralpurpose; li2022blackboxliegroup; li2024stochastichessianfittingslie; pethick2025trainingdeeplearningmodels) 提升训练效率的潜力。其中，jordan2024muon提出的 Muon 通过牛顿-舒尔茨迭代，利用正交化梯度动量更新矩阵参数。初步实验表明，Muon 在小规模语言模型训练中展现出令人鼓舞的结果。然而，正如本博客 (jordan2024muon) 所讨论的，仍有几个关键挑战亟待解决：(1) 如何有效地将基于矩阵正交化的优化器扩展至拥有数十亿参数、训练数据达数万亿 token 的大型模型，(2) 在分布式情景下如何计算近似正交化，以及 (3) 此类优化器能否在包括预训练与监督微调 (SFT) 在内的不同训练阶段间实现泛化。

在本技术报告中，我们针对这些挑战进行了全面研究。我们的工作基于 Muon，同时系统性地识别并解决了其在大规模训练场景中的局限性。我们的技术贡献包括：

- **缪子有效缩放分析**：通过广泛分析，我们发现权重衰减在缪子的可扩展性中起着关键作用。此外，我们提出了对缪子参数更新规则的缩放调整。这些调整使得缪子无需超参数调参即可开箱即用，并显著提升了训练稳定性。
- **高效分布式实现**：我们开发了 Muon 的分布式版本，采用 ZeRO-1 (Rajbhandari_2020) 风格的最优化，实现了最佳内存效率并减少了通信开销，同时保持了算法的数学特性。
- **缩放法则验证**：我们进行了缩放法则研究，将 Muon 与强大的 AdamW 基准进行比较，展示了 Muon 的卓越性能 (1a)。基于缩放法则的结果，Muon 在仅需约 52% 的训练 FLOPs 的情况下，实现了与 AdamW 训练相当的性能。

我们的全面实验表明，Muon 能够有效替代 AdamW，成为大规模 LLM 训练的实际优化器，在训练效率和模型性能上均带来显著提升。基于此项工作，我们发布了 Moonlight，一个使用 Muon 训练的 16B 参数 MoE 模型，同时提供了我们的实现及中间训练检查点，以促进 LLM 可扩展最优化技术的进一步研究。

2 方法

2.1 背景

子优化器 最近提出的 子 (jordan2024muon) 用于优化可表示为矩阵的神经网络权重。在迭代 t 时，给定当前权重 W_{t-1}、动量 μ、学习率 η_t 和目标 $\mathcal{L}_t$， 子优化器的更新规则可表述如下：

$$\begin{aligned} \mathrm{M}_t &= \mu \mathrm{M}_{t-1} + \nabla \mathcal{L}_t(\mathrm{W}_{t-1}) \\ \mathrm{O}_t &= \text{Newton-Schulz}(\mathrm{M}_t)^1 \\ \mathrm{W}_t &= \mathrm{W}_{t-1} - \eta_t \mathrm{O}_t \end{aligned} \tag{1}$$

这里，M_t 是梯度在迭代 t 时的动量，当 $t = 0$ 时设为零矩阵。在公式 1中，采用了牛顿-舒尔茨迭代过程 (bernstein2024oldoptimizernewnorm) 来近似求解 $(\mathrm{M}_t\mathrm{M}_t^{\mathrm{T}})^{-1/2}\mathrm{M}_t$ 。设 $\mathrm{U}\,\mathrm{V}^{\mathrm{T}} = \mathrm{M}_t$ 为 M_t 的奇异值分解

[1]实际上，我们遵循 (jordan2024muon)，使用 Nesterov 风格的动量，将 $\mu\mathrm{M}_t + \nabla\mathcal{L}_t(\mathrm{W}_{t-1})$ 代入牛顿-舒尔茨迭代而非 M_t。

(SVD)，我们将得到 $(\mathrm{M}_t\mathrm{M}_t^{\mathrm{T}})^{-1/2}\mathrm{M}_t = \mathrm{U}\mathrm{V}^{\mathrm{T}}$，它正交化了 M_t。直观上，正交化能确保更新矩阵是同构的，防止权重沿少数主导方向学习 (jordan2024muon)。

牛顿-舒尔茨迭代用于矩阵正交化 公式 1 通过一个迭代过程计算得出。初始时，我们设定 $\mathrm{X}_0 = \mathrm{M}_t/\|\mathrm{M}_t\|_{\mathrm{F}}$ 。随后，在每次迭代 k 中，我们根据 X_{k-1} 更新 X_k ，如下所示：

$$\mathrm{X}_k = a\mathrm{X}_{k-1} + b(\mathrm{X}_{k-1}\mathrm{X}_{k-1}^{\mathrm{T}})\mathrm{X}_{k-1} + c(\mathrm{X}_{k-1}\mathrm{X}_{k-1}^{\mathrm{T}})^2\mathrm{X}_{k-1} \tag{2}$$

其中 X_N 是经过 N 次迭代步骤后该过程的结果。这里 a、b、c 是系数。为了确保方程 2 的正确收敛，我们需要调整这些系数，使得多项式 $f(x) = ax + bx^3 + cx^5$ 在 1 附近有一个固定点。在 jordan2024muon 的原始设计中，系数被设置为 $a = 3.4445$、$b = -4.7750$、$c = 2.0315$，以便使迭代过程对于较小的初始奇异值更快收敛。在本工作中，我们遵循相同的系数设置。

范数约束下的最速下降法 bernstein2024oldoptimizernewnorm 提出将深度学习中的最优化过程视为范数约束下的最速下降法。从这一视角出发，我们可以将 Muon 与 Adam (adam2015kingma; loshchilov2018decoupled) 之间的差异视为范数约束的不同。Adam 是在动态调整的最大范数约束下的最速下降法，而 Muon 则提供了一个位于静态范围内的 Schatten-p 范数约束，适用于某些较大的 p (muoncase2024cesista)。当方程 1 被精确计算时，Muon 提供的范数约束将是谱范数。神经网络的权重被用作输入空间或隐藏空间上的算子，这些空间通常（局部）是欧几里得空间 (cesista2024firstordernormedopt)，因此对权重的范数约束应是一个诱导算子范数（或权重矩阵的谱范数）。从这个意义上讲，Muon 提供的范数约束比 Adam 提供的更为合理。

2.2 放大缪子

权重衰减 尽管 Muon 在小规模上表现显著优于 AdamW，如jordan2024muon所示，但我们发现当扩展到训练更大模型并使用更多 token 时，性能提升逐渐减弱。我们观察到，权重和层输出的 RMS 持续增长至较大值域，超出了 bf16 的高准确率范围，这可能会损害模型的性能。为解决此问题，我们将标准 AdamW（loshchilov2018decoupled）的权重衰减机制引入到 Muon 中。[2].

$$\mathrm{W}_t = \mathrm{W}_{t-1} - \eta_t(\mathrm{O}_t + \lambda\mathrm{W}_{t-1}) \tag{3}$$

我们在 Muon 上进行了有和没有权重衰减的实验，以了解其对 LLM 训练动态的影响。基于我们在第3.2节的缩放法则研究，我们训练了一个 800M 参数的模型，使用了 100B 的 token（$\sim 5\times$ 最优训练 token）。图2展示了使用 AdamW、普通 Muon（无权重衰减）和带权重衰减的 Muon 训练的模型的验证损失曲线。虽然普通 Muon 最初收敛得更快，但我们观察到一些模型权重随时间增长过大，可能限制了模型的长期性能。添加权重衰减解决了这个问题——结果表明，带权重衰减的 Muon 在过训练阶段表现优于普通 Muon 和 AdamW，实现了更低的验证损失。因此，我们将更新规则调整为公式3，其中 λ 是权重衰减比率。

一致的更新均方根 Adam 和 AdamW（adam2015kingma, loshchilov2018decoupled）的一个重要特性是它们保持理论上的更新均方根约为 1[3]. 然而，我们证明 Muon 的更新均方根值会依据以下引理随 Parameter 形状的不同而变化：

Lemma 1. *对于形状为 $[A, B]$ 的满秩矩阵参数，其理论 Muon 更新 RMS 为 $\sqrt{1/\max(A, B)}$。*

证明可在附录A中找到。我们监测了 Muon 在训练期间的更新 RMS，发现其通常接近上述理论值。我们注意到，这种不一致性在扩大模型大小时可能会带来问题。

[2]Muon 的原始实现忽略了权重衰减。最近在 Muon 中的一项并发工作引入了权重衰减，并展示了性能的提升。参见this commit和this discussion。

[3]由于 Adam 的 $\beta_1 < \beta_2$ 和 $\epsilon > 0$，实际更新的 RMS 通常小于 1。

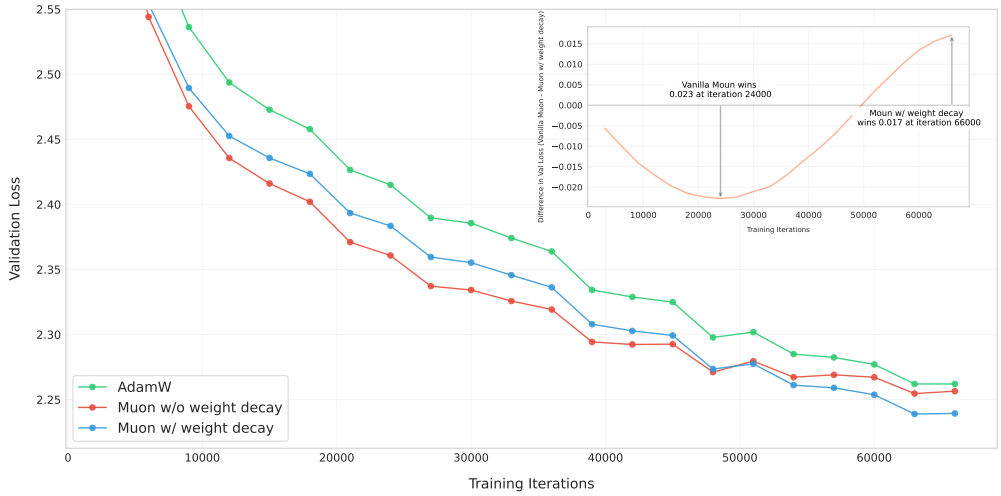


图 2: 验证损失曲线 for AdamW (green), Muon 无权重衰减 (red), 和 Muon 有权重衰减 (blue).

- 当 $\max(A, B)$ 过大时，例如稠密 MLP 矩阵，更新变得过小，从而限制了模型的表示容量，导致性能不佳；
- 当 $\max(A, B)$ 过小时，例如将 GQA (shazeer2019fasttransformerdecodingwritehead) 或 MLA (deepseekai2024deepseekv3technicalreport) 中的每个 KV 头视为分离的参数，更新会变得过大，从而导致训练不稳定，并导致性能不佳。

为了在不同形状的矩阵之间保持一致的更新 RMS，我们建议通过每个矩阵的 $\sqrt{\max(A, B)}$ 来缩放 Muon 更新，以消除引理 1的影响。[4] . 第 3.1 节中的实验表明，该策略对最优化是有益的。

AdamW 的 RMS 匹配更新 Muon 旨在更新基于矩阵的参数。实际应用中，AdamW 与 Muon 配合使用，以处理非矩阵参数，如 RMSNorm、LM 头部和嵌入参数。我们希望优化器超参数（学习率 η，权重衰减 λ）在矩阵和非矩阵参数之间共享。

我们建议将 Muon 的更新均方根（RMS）调整至与 AdamW 相近的水平。根据经验观察，AdamW 的更新 RMS 通常位于 0.2 至 0.4 之间。因此，我们通过以下调整将 Muon 的更新 RMS 缩放至这一值域：

$$\mathrm{W}_t = \mathrm{W}_{t-1} - \eta_t(0.2 \cdot \mathrm{O}_t \cdot \sqrt{\max(A, B)} + \lambda \mathrm{W}_{t-1}) \tag{4}$$

我们通过实证结果验证了这一选择（详见附录A）。此外，我们强调，通过这一调整，Muon 可以直接**复用**为 AdamW 调整好的学习率和权重衰减。

其他超参数 Muon 包含两个可调节的超参数：Newton-Schulz 迭代步数和动量 μ。我们经验性地观察到，当将 N 设置为 10 时，迭代过程将产生比 $N = 5$ 更精确的正交化结果，但这并不会带来更好的性能。因此，为了效率起见，我们在这项工作中将 $N = 5$ 设置为默认值。在调整动量时，我们没有看到一致的性能提升，因此我们选择了与 jordan2024muon 相同的 0.95。

[4] jordan2024muon 的原始实现通过 $\sqrt{\max(1, A/B)}$ 对更新进行缩放，如果所有矩阵的第二维度相同，则相当于我们的提议（除了一个全局缩放因子）；pethick2025trainingdeeplearningmodels 和 JiachengX 在我们的工作同时讨论了关于更新缩放因子的类似问题。

Algorithm 1 Distributed Muon

Require: Full Gradients G, DP partitioned Momentum m, DP partitioned parameters p, momentum μ.

// Reduce-scatter G on DP for correct gradients
g = reduce_scatter(G, dp_group)
// Apply momentum to g using local partitioned momentum m
$g' = \text{update\_with\_momentum}(g, m, \mu)$
// DP Gather: gathering g' across DP into a full matrix G
G = gather(g', dp_group)
// Calculate Muon update
U = Newton-Schulz(G)
// Discard the rest of U and only keep the local partition u, then apply the update rule
$p' = \text{apply\_update}(p, u)$
// All-gather updated p' into P
P = all_gather(p', dp_group)
// Return the update RMS for logging
return $\sqrt{u^2.\texttt{mean}()}$

2.3 分布式缪子

ZeRO-1 与 Megatron-LM Rajbhandari_2020 提出了 ZeRO-1 技术，该技术将昂贵的优化器状态（如主权重、动量）在整个集群中进行划分。Megatron-LM (shoeybi2020megatronlmtrainingmultibillionparameter) 将 ZeRO-1 集成到其原生并行设计中。基于 Megatron-LM 的复杂并行策略，例如张量并行（TP）、流水线并行（PP）、专家并行（EP）和数据并行（DP），ZeRO-1 的通信负载可以从在整个分布式环境中收集减少到仅在数据并行组内收集。

方法 ZeRO-1 对 AdamW 有效，因其以逐元素方式计算更新。然而，Muon 需要完整的梯度矩阵来计算更新。因此，原始的 ZeRO-1 并不直接适用于 Muon。我们提出了一种基于 ZeRO-1 的新分布式解决方案，称为分布式 Muon。分布式 Muon 遵循 ZeRO-1 在数据并行（DP）上划分优化器状态，并相较于原始的 ZeRO-1 AdamW 优化器引入了两项额外操作：

1. `DP Gather`. 对于本地 DP 划分的主权重（$1/DP$ 模型权重的大小），此操作将对应的划分梯度收集到一个完整的梯度矩阵中。
2. 计算完整更新。在上述收集步骤之后，按照第 2.1 节所述，对完整梯度矩阵执行牛顿-舒尔茨迭代。需要注意的是，随后我们将丢弃完整更新矩阵的一部分，因为我们只需要与本地参数对应的划分来执行更新。

分布式 Muon 的实现描述见算法1。分布式 Muon 引入的额外操作以蓝色标注。

分析 我们将分布式 Muon 与基于经典 ZeRO-1 的分布式 AdamW（简称为分布式 AdamW）在多个方面进行了比较：

- 内存使用。 Muon 仅使用一个动量缓冲区，而 AdamW 使用两个动量缓冲区。因此，Muon 优化器所使用的额外内存仅为分布式 AdamW 的一半。
- 通信开销。对于每个设备，额外的 DP 收集仅需针对本地 DP 划分的参数 p。因此，通信成本低于 G 的 reduce-scatter 或 P 的 all-gather。此外，Muon 仅需在 bf16 中执行 Newton-Schulz 迭代步骤，从而将通信开销进一步减少至 fp32 的 50%。总体而言，分布式 Muon 的通信工作量为分布式 AdamW 的 $(1, 1.25]$。上限计算为：分布式 Muon 的通信为 4（fp32 G reduce-scatter）+ 2（bf16 Muon gather）+ 4（fp32 P

all-gather)，而分布式 AdamW 为 4 + 4。在实际应用中，由于我们通常使用多个 DP 进行训练，经验上的额外成本通常更接近下限 1。[5].

- 延迟。分布式 Muon 的端到端延迟比分布式 AdamW 更大，因为它引入了额外的通信并需要运行 Newton-Schulz 迭代步骤。然而，这并不是一个显著问题，因为 (a) 仅需约 5 次 Newton-Schulz 迭代步骤即可获得良好结果（如第2.2节所述），且 (b) 由优化器引起的端到端延迟相较于模型的前向-反向传播时间（通常为 1% 至 3%）可以忽略不计。此外，多种工程技巧，如重叠收集与计算，以及重叠优化器的 reduce-scatter 与参数收集，可以进一步减少延迟。

在分布式簇中训练大规模模型时，Distributed Muon 相比其 AdamW 对应方案没有明显的延迟开销。我们即将发布一个拉取请求，为开源项目 Megatron-LM (shoeybi2020megatronlmtrainingmultibillionparameter) 实现 Distributed Muon。

3 实验

3.1 一致更新均方根

如第 2.2 节所述，我们的目标是使所有矩阵参数的更新 RMS 相匹配，并与 AdamW 的 RMS 保持一致。我们尝试了两种方法来控制参数间的 Muon 更新 RMS，并将其与仅保持与 AdamW 一致的 RMS 的基准进行了比较：

1. 基准。我们将更新矩阵乘以 $0.2 \cdot \sqrt{H}$（H 为模型隐藏层大小），以保持与 AdamW 一致的更新均方根。注意，对于大多数矩阵，$\max(A, B)$ 等于 H。
$$\mathrm{W}_t = \mathrm{W}_{t-1} - \eta_t(0.2 \cdot \mathrm{O}_t \cdot \sqrt{H} + \lambda \mathrm{W}_{t-1}) \tag{5}$$
2. 更新范数。我们可以直接对通过 Newton-Schulz 迭代计算出的更新进行归一化，使其 RMS 严格变为 0.2；
$$\mathrm{W}_t = \mathrm{W}_{t-1} - \eta_t(0.2 \cdot \mathrm{O}_t / \operatorname{RMS}(\mathrm{O}_t) + \lambda \mathrm{W}_{t-1}) \tag{6}$$
3. 调整后的学习率。对于每个更新矩阵，我们可以根据其形状通过一个因子 $0.2 \cdot \sqrt{\max(A, B)}$ 来缩放其学习率。
$$\mathrm{W}_t = \mathrm{W}_{t-1} - \eta_t(0.2 \cdot \mathrm{O}_t \cdot \sqrt{\max(A, B)} + \lambda \mathrm{W}_{t-1}) \tag{7}$$

分析 我们设计了实验以展示 Muon 更新 RMS 在早期训练阶段的影响，因为我们在更大规模模型训练中观察到异常行为迅速出现。我们按照3.2所述，对 800M 小规模模型进行了实验。当矩阵维度差异增大时，更新 RMS 不一致的问题更为显著。为了突出问题以供进一步研究，我们略微调整了模型架构，将 Swiglu MLP 替换为标准的两层 MLP，将其矩阵参数形状从 $[H, 2.6H]$ 改为 $[H, 4H]$。我们评估了模型的损失，并监控了部分参数的 RMS，特别是注意力查询（形状 $[H, H]$）和 MLP（形状 $[H, 4H]$）。在完成 20B token 训练计划中的 4B token 后，我们对模型进行了评估。从表 1中，我们观察到了几个有趣的发现：

1. `Update Norm` 和 `Adjusted LR` 均取得了优于 `Baseline` 的表现；
2. 对于形状为 $[H, 4H]$ 的 MLP 权重矩阵，更新范数和调整学习率获得的权重 RMS 大约是基准的两倍。这是合理的，因为 $\sqrt{\max(H, 4H)}/\sqrt{H} = 2$，所以更新范数和调整学习率的更新 RMS 大约是基准的两倍；
3. 对于形状为 $[H, H]$ 的注意力查询权重矩阵，`Update Norm` 仍然对更新进行范数处理，而 `Adjusted LR` 则不会，因为 $\sqrt{\max(H, H)}/\sqrt{H} = 1$。因此，`Adjusted LR` 的权重 RMS 与 `Baseline` 相似，但 `Update Norm` 的权重 RMS 较大，类似于其 MLP。

基于这些发现，我们选择 `Adjusted LR` 方法用于未来实验，因为其成本较低。

[5]若启用了 TP，分布式 Muon 需要在 TP 组上额外执行一次 bf16 TP 聚集操作。

表 1: 控制不同模型参数下 子的更新均方根值

Methods	Training loss	Validation loss	query weight RMS	MLP weight RMS
Baseline	2.734	2.812	3.586e-2	2.52e-2
Update Norm	2.72	2.789	4.918e-2	5.01e-2
Adjusted LR	2.721	2.789	3.496e-2	4.89e-2

表 2: 缩放法则模型与超参数

# Params. w/o Embedding	Head	Layer	Hidden	Tokens	LR	Batch Size*
399M	12	12	1536	8.92B	9.503e-4	96
545M	14	14	1792	14.04B	9.143e-4	128
822M	16	16	2048	20.76B	8.825e-4	160
1.1B	18	18	2304	28.54B	8.561e-4	192
1.5B	20	20	2560	38.91B	8.305e-4	256

*In terms of number of examples in 8K context length.

3.2 缪子缩放法则

为了与 AdamW 进行公平比较，我们在 Llama (grattafiori2024llama3herdmodels) 架构下对一系列稠密模型进行了缩放法则实验。在优化器研究中，建立一个强大的基准至关重要。因此，我们按照计算最优的训练设置 (kaplan2020scalinglawsneurallanguage)(the grid search experiments can be found in Appendix B)，对 AdamW 的超参数进行了网格搜索。模型架构和超参数的详细信息可在表 2中找到。对于 Muon，如第 2.2节所述，由于我们将 Muon 的更新 RMS 与 AdamW 相匹配，因此直接重用了对 AdamW 基准最优的超参数。

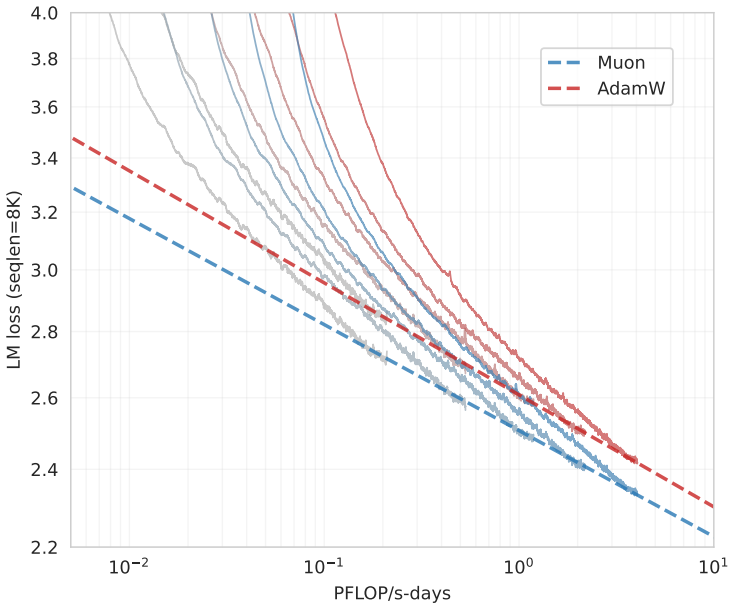


图 3: 拟合的 Muon 与 AdamW 优化器的缩放法则曲线。

拟合的缩放法则曲线可以在图3中找到，拟合方程详见表3。如图 1a所示，在计算最优情景下，Muon 仅需约 52% 的训练 FLOPs 即可达到与 AdamW 相当的性能。

表 3: 缩放法则曲线的拟合参数

	Muon	AdamW
LM loss (seqlen=8K)	$2.506 \times C^{-0.052}$	$2.608 \times C^{-0.054}$

3.3 使用 Muon 进行预训练

模型架构 为了评估 Muon 与当代模型架构的对比，我们从头开始预训练，采用了 deepseek-v3-small 架构 (deepseekai2024deepseekv3technicalreport)，因为它展示了强大的性能，并且原始结果可作为比较的参考。我们的预训练模型拥有 24 亿激活参数和 152.9 亿总参数（包含嵌入时，激活参数为 30 亿，总参数为 160 亿）。对架构的细微修改详见附录 C。

预训练数据 我们的预训练数据详情可在k1p5中找到。预训练期间的最大上下文长度为 8K。

预训练 该模型分多个阶段进行训练。在第一和第二阶段，我们使用 1e-3 的 auxfree 偏置更新率，而在第三阶段则设为 0.0。所有阶段的权重衰减均设置为 0.1。更多关于模型训练的细节和讨论可参见附录D。

1. `0` 到 `33B token`：在此阶段，学习率在 2k 步内线性增加至 4.2e-4。批量大小保持在 2048 个样本；
2. `33B` 到 `5.2T token`：在此阶段，学习率以余弦方式从 4.2e-4 衰减至 4.2e-5。我们将批量大小保持在 2048，直到处理完 200B token，随后将其翻倍至 4096 继续处理剩余部分；
3. `5.2T` 到 `5.7T tokens`：在此阶段（也称为冷却阶段），学习率在 100 步内增加到 1e-4，然后在 500B tokens 内线性衰减至 0，并保持 4096 的批量大小不变。在此阶段，我们使用最高质量的数据，重点关注数学、代码和推理。

评估基准 我们的评估涵盖了四大类基准，每类基准旨在评估模型的不同能力：

- **英语语言理解与推理**：MMLU（5-shot）(hendrycks2021measuringmassivemultitasklanguage)，MMLU-pro（5-shot）(wang2024mmluprorobustchallengingmultitask)，BBH（3-shot）(suzgun2022challengingbigbenchtaskschainofthought)，TriviaQA（5-shot）(joshi2017triviaqalargescaledistantly)
- **代码生成**：HumanEval(通过@1) (chen2021codex)，MBPP(通过@1)(austin2021programsynthesislargelanguage)
- **数学推理**：GSM8K（4-shot）(cobbe2021trainingverifierssolvemath) MATH (hendrycks2021measuringmathematicalproblemsolving), CMATH (wei2023cmathlanguagemodelpass)
- **中文语言理解与推理能力评估**：C-Eval（5-shot）(huang2023cevalmultilevelmultidisciplinechinese)，CMMLU（5-shot）(li2024cmmlumeasuringmassivemultitask)

性能 我们将使用 Muon 训练的模型命名为 “Moonlight”。我们将 Moonlight 与不同公开的模型在相似规模上进行了比较。我们首先在 1.2T token 上评估了 Moonlight，并将其与以下具有相同架构且训练 token 数量相当的模型进行了对比：

- `Deepseek-v3-Small` (deepseekai2024deepseekv3technicalreport) 是一个拥有 2.4B/16B 参数的 MoE 模型，使用 1.33T token 进行训练；
- `Moonlight-A` 遵循与 Moonlight 相同的训练情景，不同之处在于它使用了 AdamW 优化器。

表 4: 不同模型在约 1.2T token 上的比较。

	Benchmark (Metric)	DSV3-Small	Moonlight-A@1.2T	Moonlight@1.2T
	Activated Params†	2.24B	2.24B	2.24B
	Total Params†	15.29B	15.29B	15.29B
	Training Tokens	1.33T	1.2T	1.2T
	Optimizer	AdamW	AdamW	Muon
English	MMLU	53.3	60.2	60.4
	MMLU-pro	-	26.8	28.1
	BBH	41.4	45.3	43.2
	TriviaQA	-	57.4	58.1
Code	HumanEval	26.8	29.3	37.2
	MBPP	36.8	49.2	52.9
Math	GSM8K	31.4	43.8	45.0
	MATH	10.7	16.1	19.8
	CMath	-	57.8	60.2
Chinese	C-Eval	-	57.2	59.9
	CMMLU	-	58.2	58.8

† The reported parameter counts exclude the embedding parameters.

对于 Moonlight 和 Moonlight-A，我们采用了总计 5.7T 预训练中的中间 1.2T token 检查点，此时学习率尚未衰减至极小值，模型也尚未经历冷却阶段。

如表4所示，我们采用 AdamW 训练的基准模型 Moonlight-A 在与类似公开模型的对比中展现了强劲的性能。Moonlight 的表现显著优于 Moonlight-A，证明了 Muon 的缩放有效性。我们观察到 Muon 尤其在数学和代码相关任务上表现卓越，鼓励研究界进一步探究这一现象。在 Moonlight 完成对 5.7T token 的全面训练后，我们将其与规模相近的公开模型进行了对比，并将结果展示在表5中：

- 来自 grattafiori2024llama3herdmodels 的 `LLAMA3-3B` 是一个拥有 3B 参数的稠密模型，使用了 9T token 进行训练。
- `Qwen2.5-3B` 来自 qwen2.5 是一个拥有 3B 参数的稠密模型，使用 18T token 进行训练。
- `Deepseek-v2-Lite` 来自 deepseekv2，是一个拥有 2.4B/16B 参数的 MoE 模型，训练使用了 5.7T 的 token。

如表 5所示，Moonlight 在相同架构和训练 token 数量的情况下表现优于其他模型。即使与在更大数据集上训练的稠密模型相比，Moonlight 仍保持竞争力。详细对比见附录 E。Moonlight 的性能还在 MMLU 和 GSM8k 上与其他知名语言模型进行了进一步比较，如图 1b及附录 E图 8所示。[6]. 值得注意的是，Moonlight 位于模型性能与训练预算的帕累托前沿，在不同大小上均优于众多其他模型。

3.4 奇异谱动力学

为了验证 Muon 能够在更多样化的方向上优化权重矩阵的直觉，我们对使用 Muon 和 AdamW 训练的权重矩阵进行了谱分析。对于一个具有奇异值 $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_n)$ 的权重矩阵，我们按如下方式计算该矩阵的

[6]性能指标和计算需求（FLOPs）的基准模型数据来源于 (olmo20242)。

表 5: 不同模型在多个基准上的比较。

	Benchmark (Metric)	Llama3.2-3B	Qwen2.5-3B	DSV2-Lite	Moonlight
	Activated Param†	2.81B	2.77B	2.24B	2.24B
	Total Params†	2.81B	2.77B	15.29B	15.29B
	Training Tokens	9T	18T	5.7T	5.7T
	Optimizer	AdamW	Unknown	AdamW	Muon
English	MMLU	54.7	65.6	58.3	70.0
	MMLU-pro	25.0	34.6	25.5	42.4
	BBH	46.8	56.3	44.1	65.2
	TriviaQA‡	59.6	51.1	65.1	66.3
Code	HumanEval	28.0	42.1	29.9	48.1
	MBPP	48.7	57.1	43.2	63.8
Math	GSM8K	34.0	79.1	41.1	77.4
	MATH	8.5	42.6	17.1	45.3
	CMath	-	80.0	58.4	81.1
Chinese	C-Eval	-	75.0	60.3	77.2
	CMMLU	-	75.0	64.3	78.2

† The reported parameter counts exclude the embedding parameters.‡ We tested all listed models with the full set of TriviaQA.

SVD 熵 (svd_entropy; effectiverank)：

$$H(\sigma) = -\frac{1}{\log n}\sum_{i=1}^{n}\frac{\sigma_i^2}{\sum_{j=1}^{n}\sigma_j^2}\log\frac{\sigma_i^2}{\sum_{j=1}^{n}\sigma_j^2}$$

如图 4所示，我们可视化了在 1.2T token 预训练过程中，不同训练检查点的权重矩阵的平均 SVD 熵。可以看出，在所有训练检查点和所有权重矩阵组中，Muon 的 SVD 熵均高于 AdamW，这验证了 Muon 能为权重矩阵提供更多样化更新谱的直觉。这种差异在专家选择的路由器权重中更为显著，表明专家混合模型能从 Muon 中获益更多。

此外，我们可视化了在训练了 1.2T token 的检查点处每个权重矩阵的奇异值分布，如附录 F所示。我们发现，对于超过 90% 的权重矩阵，Muon 优化后的 SVD 熵高于 AdamW，这为 Muon 在探索多样化最优化方向上的卓越能力提供了强有力的实证证据。

3.5 监督微调（SFT）与 Muon

在本节中，我们对 LLM 训练标准 SFT 阶段内的 Muon 优化器进行了消融研究。研究结果表明，Muon 在 SFT 阶段带来的优势依然显著。具体而言，在消融实验中，同时经过 Muon 预训练和 Muon 微调的模型表现优于其他模型。然而，我们也观察到，当 SFT 优化器与预训练优化器不同时，使用 Muon 进行 SFT 相较于 AdamW 并未展现出明显优势。这表明仍有较大的探索空间，我们将其留待未来工作进一步研究。

3.5.1 预训练与 SFT 优化器互换性的消融研究

为了进一步探究 Muon 的潜力，我们使用 Muon 和 AdamW 优化器对 Moonlight@1.2T 和 Moonlight-A@1.2T 进行了微调。这些模型在包含 4k 序列长度的开源 tulu-3-sft-mixture 数据集 (lambert2024tulu3) 上进行了两轮次的微调。学习率采用线性衰减策略，从 5×10^{-5} 开始，逐步降低至 0。表6中的结果显示，Moonlight@1.2T 相较于 Moonlight-A@1.2T 表现更为优异。

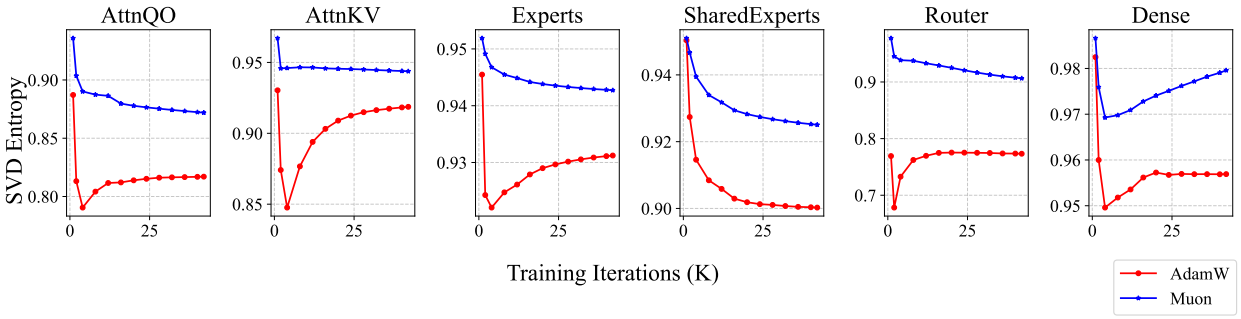


图 4: 不同训练迭代中权重矩阵的奇异值分解熵。我们将权重矩阵分为六类：1) AttnQO 代表与注意力层中查询及输出投影相关的权重矩阵；2) AttnKV 代表与注意力层中键值投影相关的权重矩阵；3) Experts 指专家模型中的权重矩阵；4) SharedExperts 指共享专家模型中的权重矩阵；5) Router 为路由器中的权重矩阵；6) Dense 是第一稠密层中的权重矩阵。奇异值分解熵计算为各层所有该类权重矩阵的宏观平均值。对于专家模型中的权重，出于效率考虑，我们仅在不同层中选取 64 个专家中的 3 个进行计算。

表 6: 研究优化器在预训练与 SFT 阶段互换使用的影响。

Benchmark (Metric)	# Shots	Moonlight-1.2T			
Pretraining Optimizer	-	Muon	AdamW	Muon	AdamW
SFT Optimzier	-	Muon	Muon	AdamW	AdamW
MMLU (EM)	0-shot (CoT)	55.7	55.3	50.2	52.0
HumanEval (Pass@1)	0-shot	57.3	53.7	52.4	53.1
MBPP (Pass@1)	0-shot	55.6	55.5	55.2	55.2
GSM8K (EM)	5-shot	68.0	62.1	64.9	64.6

3.5.2 SFT 与 Muon 在公开预训练模型上的应用

我们进一步将 Muon 应用于一个公开预训练模型的监督微调(SFT)，具体是 Qwen2.5-7B 基础模型(qwen2.5)，使用了开源的 tulu-3-sft-mixture 数据集 (lambert2024tulu3)。该数据集以 8k 序列长度打包，并采用了余弦衰减学习率调度，起始于 2×10^{-5}，逐步降至 2×10^{-6}。结果展示在表7中。为作比较，我们展示了经 Muon 微调的模型表现与 Adam 微调模型相当。这些结果表明，为达到最佳性能，在预训练阶段应用 Muon 比在监督微调阶段更为有效。

表 7: 应用于 Qwen2.5-7B 预训练模型 SFT 的 Adam 与 Muon 优化器对比。

Benchmark (Metric)	# Shots	Adam-SFT	Muon-SFT
Pretrained Model	-	Qwen2.5-7B	
MMLU (EM)	0-shot (CoT)	71.4	70.8
HumanEval (Pass@1)	0-shot	79.3	77.4
MBPP (Pass@1)	0-shot	71.9	71.6
GSM8K (EM)	5-shot	89.8	85.8

4 讨论

未来研究可沿多个方向深入探索并拓展当前的研究成果。

将全部参数整合至 Muon 框架中 目前，Muon 优化器与 Adam 优化器联合使用，其中部分参数仍由 Adam 优化负责。这种混合方法虽可行，但存在改进空间。将所有参数的最优化完全纳入 Muon 框架内，是一个备受关注的研究主题。

将 Muon 扩展至 Schatten 范数 Muon 优化器可视为在谱范数下的最速下降法。鉴于 Schatten 范数的广泛适用性和多功能性，将 Muon 扩展至涵盖一般 Schatten 范数是一个极具前景的方向。这一扩展可能解锁额外的优化能力，并有望在现有基于谱范数的实现基础上取得更优的结果。

理解与解决预训练-微调不匹配问题 实践中观察到一个显著现象，即使用 AdamW 预训练的模型在使用 Muon 微调时表现不佳，反之亦然。这种优化器不匹配现象严重阻碍了有效利用大量 AdamW 预训练检查点的潜力，因此需要进行严格的理论研究。深入理解其潜在机制对于设计出稳健且有效的解决方案至关重要。

5 结论

在本技术报告中，我们对 Muon 在 LLM 训练中的可扩展性进行了全面研究。通过系统分析和改进，我们成功将 Muon 应用于一个基于 5.7 万亿 token 训练的 3B/16B 参数 MoE 模型。我们的结果表明，Muon 能够有效替代 AdamW 作为大规模 LLM 训练的标准优化器，在训练效率和模型性能方面均展现出显著优势。通过开源我们的实现、Moonlight 模型以及中间训练检查点，我们旨在促进可扩展最优化技术的进一步研究，并加速 LLM 训练方法的开发。

A 更新 RMS

引理 1 的证明

证明. 在不失一般性的情况下，考虑正交矩阵 $U \in \mathbb{R}^{n\times n}$ 和 $V \in \mathbb{R}^{m\times m}$，其中 $n \geq m \geq r$。我们将证明对于 $X = U_{[:,:r]}V_{[:r,:]}$（Muon 的更新具有相同的格式），RMS 值为 $\sqrt{r/mn}$。根据矩阵乘法的定义：

$$X_{i,j} = \sum_{k=1}^{r} U_{i,k}V_{k,j}$$

均方根（RMS）可表示为：

$$\begin{aligned}
\mathrm{RMS}(X)^2 &= \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r} U_{i,k}^2 V_{k,j}^2 \\
&= \frac{1}{mn}\sum_{k=1}^{r}\left(\sum_{i=1}^{n} U_{i,k}^2\right)\left(\sum_{j=1}^{m} V_{k,j}^2\right) \\
&= \frac{1}{mn}\sum_{k=1}^{r} 1 \\
&= \frac{r}{mn}
\end{aligned}$$

因此，$\mathrm{RMS}(X) = \sqrt{r/mn}$。对于矩阵满秩的常见情况，$r = m$，从而得到 $\mathrm{RMS}(X) = \sqrt{1/n}$。 □

在 Muon 与 AdamW 优化器间保持一致的更新 RMS 如2.2所述，我们希望匹配 Muon 与 AdamW 优化器的更新 RMS。这一目标通过小规模模型实验得到验证。我们将 Muon 的更新 RMS 设定在 $[0.05, 0.1, 0.2, 0.4, 0.8]$ 范围内，并以 AdamW 作为基准。在表8中，我们报告了在 2 千步（约 20 亿 token）时的损失及代表性权重矩阵的 RMS。结果显示，0.2 RMS 与 0.4 RMS 表现相近且显著优于其他情景。这些发现与我们经验观察一致，即 AdamW 的更新 RMS 位于 $0.2 \sim 0.4$ 范围内。因此，我们选择将 Muon 的更新 RMS 控制在 0.2。

表 8: 子更新均方根实验

Optimizer	AdamW	0.05 RMS*	0.1 RMS	0.2 RMS	0.4 RMS	0.8 RMS
LM training loss	3.512	3.355	3.239	3.198	3.199	3.386
LM validation loss	3.679	3.503	3.374	3.325	3.314	3.543
AttnQ weight RMS	1.01e-2	5.74e-3	8.44e-3	1.57e-2	2.95e-2	7.23e-2
Mlp weight RMS	1.25e-2	8.01e-3	1.27e-2	2.35e-2	4.51e-2	8.73e-2

*Except the first column, all other candidates are using Muon with controlled RMS.

B AdamW 基准缩放法则

为确保实验的公平性与准确率，我们在自有数据集上进行了一系列实验，以得出适用于 AdamW 的最优缩放法则参数。这包括在受限的计算预算（FLOPs，C）下确定最优模型大小 (N)、训练 token 数量 (D)、学习率 (η) 及批量大小 (B)。 (kaplan2020scalinglawsneurallanguage; hoffmann2022trainingcomputeoptimallargelanguage; bi2024deepseek) 表9展示了我们系统参数挖掘过程的结果。

表 9: 缩放法则参数与计算预算（FLOPs）之间的经验关系

$N(C)$	$D(C)$	$\eta(C)$	$B(C)$
$0.0483359 \cdot C^{0.5112684}$	$3.4480927 \cdot C^{0.4887316}$	$0.0127339 \cdot C^{-0.0574752}$	$0.0065202 \cdot C^{0.4137915}$

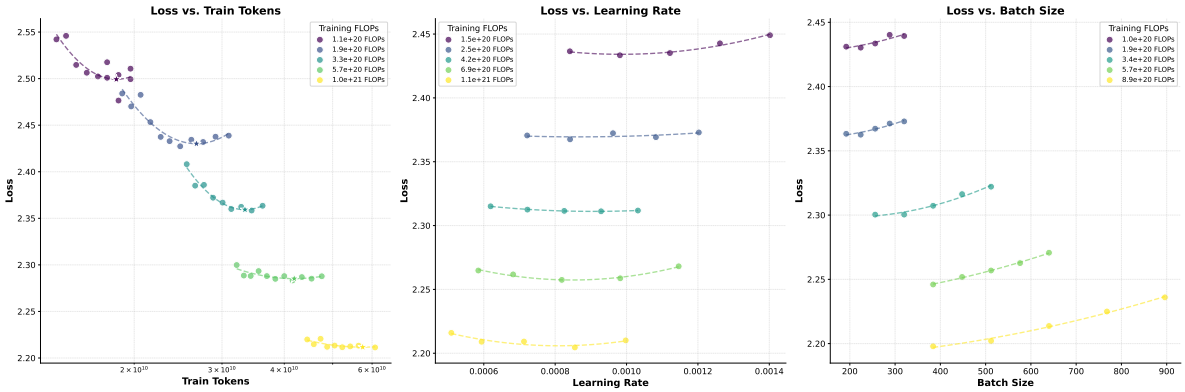


图 5: 跨 FLOP 预算的缩放法则超参数优化地形

超参数搜索 为了系统性地识别 AdamW 基准中的最优缩放法则超参数，我们采用了多阶段搜索协议。首先，我们选择了多个计算预算（FLOPs 水平）并根据先前研究的经验指南初始化了模型大小、学习率和批量大小。对于每个固定的 FLOPs 约束，我们改变了模型大小 N，同时调整训练 token 数量 D 以保持 $C = 6ND$，从而探索模型容量与数据效率之间的权衡。每个配置都训练至收敛，并记录验证损失以确定 N 和 D 的帕累托最优组合。随后，在固定最优 $N - D$ 对的基础上，我们通过网格搜索进一步优化学习率和批量大小，确保配置间的稳定性和收敛性。为了缓解局部极小值并增强鲁棒性，这一迭代过程重复了 2-3 次，逐步缩小了超参数空间。

最优化过程在图5中进一步展示，该图描绘了在不同 FLOPs 预算下，损失函数随训练 token、学习率和批量大小变化的景观。每个碗状曲线代表特定 FLOPs 水平下的损失表面，其独特的全局最小点对应着最优的超参数配置。

C 模型架构

Muon 对模型架构保持中立，我们采用了与 DeepSeek-V3-Small 相似的模型，如deepseekai2024deepseekv3technicalreport所述，因其作为基准模型，具有开放权重且性能强劲。在 Moonlight 模型中，我们进行了几处细微调整，并在此列出：

多 Token 预测 (MTP) 在我们的实验中，MTP 并未显示出对预训练的显著益处。为简化起见，我们未将 MTP 层引入 Moonlight 模型。

无辅助偏置更新 在 deepseekai2024deepseekv3technicalreport 中，无辅助偏置通过以下方式更新：$b_i = b_i + u \times \text{sign}(e_i)$，其中 u 为更新比率，b_i 为第 i 个专家的偏置，e_i 为专家的违反比率。我们略微修改了更新规则为：$b_i = b_i + u \times (\text{sign}(e_i) - \text{sign}(e).\text{mean}())$，其中 $\text{sign}(e).\text{mean}()$ 为所有专家违反比率符号的平均值，以控制偏置的幅度，同时不改变 topk 选择逻辑。

门控缩放因子 Deepseek-V2-Lite 未采用门控缩放因子，而 Deepseek-V3 使用了 2.5 的缩放因子。我们采用了 2.446 的缩放因子，以控制与稠密模型相似的输出均方根。计算我们门控缩放因子的代码可在图 6 中找到。

```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def calc_gate_scaling_factor(num_experts: int, topk: int, iter_times: int):
    """Calculate the gate scaling factor for MoE.

    Args:
        num_experts (int): The number of experts.
        topk (int): The number of experts to select.
        iter_timers (int): The number of iterations.

    Returns:
        float: The gate scaling factor.
    """
    factors = []
    for _ in range(iter_times):

        # mock gaussian logits
        logits = np.random.randn(num_experts)
        # select topk logits
        p = np.sort(sigmoid(logits))[::-1]
        p = p[:topk]
        # renormalize
        p = p / p.sum()
        # calculate the scaling factor
        factors.append( 1/ (p**2).sum()**0.5)
    return np.mean(factors)
```

图 6: 计算门缩放因子的 Python 实现。

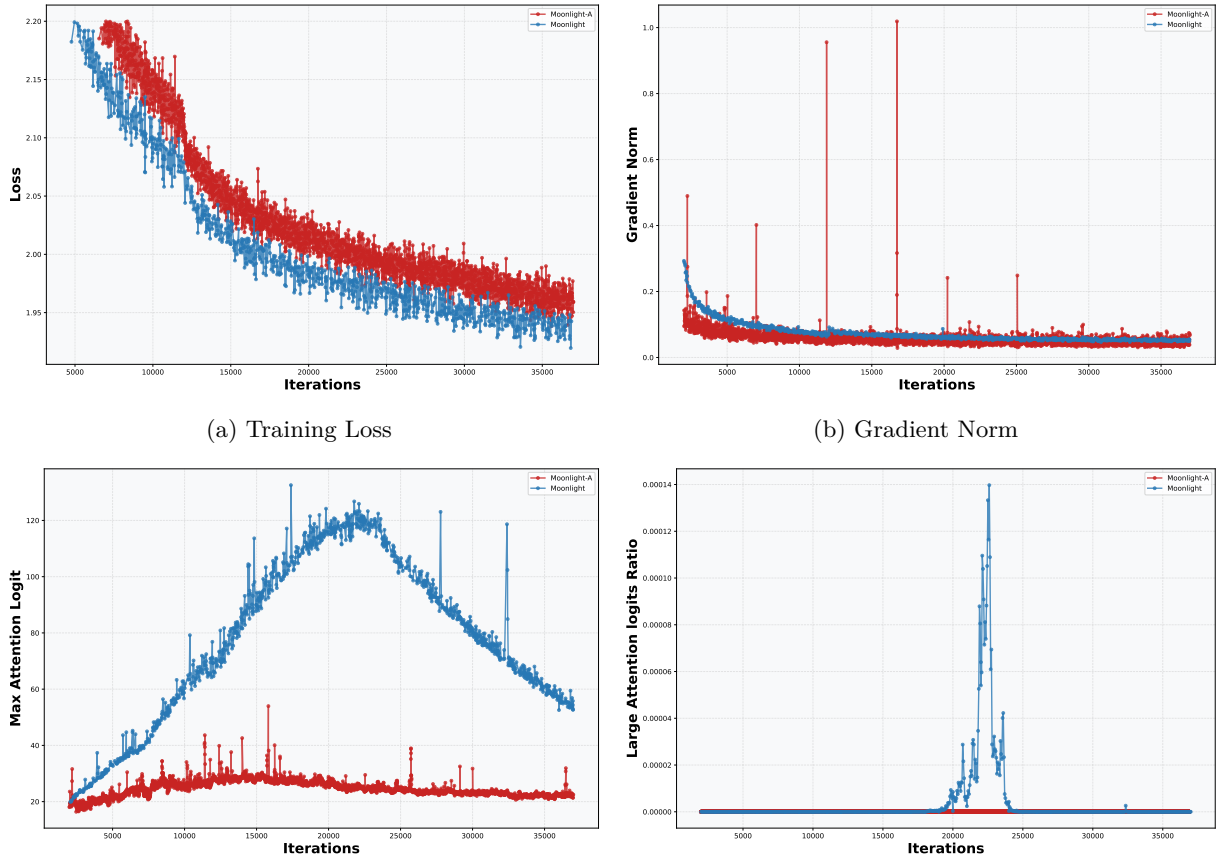
D 训练稳定性

无损失或梯度范数尖峰 Moonlight 的训练过程非常顺利，我们没有遇到任何损失尖峰或梯度范数尖峰。损失和梯度范数曲线如图7所示（Moonlight 以蓝色表示，由 AdamW 训练的 Moonlight-A 以红色表示）。

在训练过程中，我们观察到尽管训练损失和梯度范数在整个过程中保持稳定，但最大注意力 Logit（计算为全局批量中的单个最大 Logit 值）在初始训练阶段的特定层中表现出明显的上升轨迹，超过了 100 的阈值。值得注意的是，与其他优化器相比，AdamW 在控制这一指标方面表现出更健康的行为。

为了进一步探究这一现象的影响，我们引入了大注意力 Logit 比例这一指标，定义为批量中超过 100 的注意力 Logit 所占比例。如图7所示，该比例始终保持在较低水平（约 10^{-4}），表明极端大的 Logit 值较为稀疏。此外，随着训练的进行，最大 Logit 值逐渐减小，暗示最优化动态趋于健康。

RMSNorm Gamma **权重衰减** 值得注意的是，对 RMSNorm gamma 参数应用权重衰减对于确保训练稳定性至关重要，因为它能有效防止每层输出 RMS 值过高。



(a) Training Loss

(b) Gradient Norm

(c) Max Attention Logit (Layer 1)

(d) Large Attention Logits Ratio (Layer 1)

图 7: 月光与月光-A 之间的训练动态对比

E 与更昂贵模型的对比

表 10 展示了我们的 Moonlight 模型（通过 Muon 优化）与使用更多计算资源训练的公开可用模型之间的对比分析，包括 `LLama3.1-8B` (grattafiori2024llama3herdmodels)、`Gemma-9B` (team2024gemma) 和 `Qwen2.5-7B` (qwen2.5)。图 8 展示了 Moonlight 在 GSM8k 基准测试中与领域内可比模型的性能对比。

F 权重矩阵的奇异值分布

我们通过为每个矩阵绘制其按降序排列的奇异值折线图来可视化权重矩阵的奇异值分布，这些值均以最大奇异值为基准进行了规范化。如图9和10所示，我们发现，对于大多数权重矩阵而言，经 Muon 优化后的奇异值分布比 AdamW 更为平坦，这进一步证实了 Muon 能够提供更丰富更新谱的假设。

表 10: 不同模型在多个基准上的比较。

	Benchmark (Metric)	Moonlight	LLAMA3.1-8B	Gemma2-9B	Qwen2.5-7B
			Larger Training Compute Model		
	Activated Param†	2.24B	7.38B	8.32B	6.83B
	Total Params†	15.29B	7.38B	8.32B	6.83B
	Training Tokens	5.7T	15T	8T	18T
	Optimizer	Muon	AdamW	Unknown	Unknown
English	MMLU	70.0	66.7	71.3	74.2
	MMLU-pro	42.4	37.1	44.7	45.0
	BBH	65.2	57.7	68.2	70.4
	TriviaQA‡	66.3	70.3	-	60.0
Code	HumanEval	48.1	37.2	37.8	57.9
	MBPP	63.8	47.6	62.2	74.9
Math	GSM8K	77.4	57.2	70.7	85.4
	MATH	45.3	20.3	37.7	49.8

† The reported parameter counts exclude the embedding parameters.‡ We test all listed models with the full set of TriviaQA.

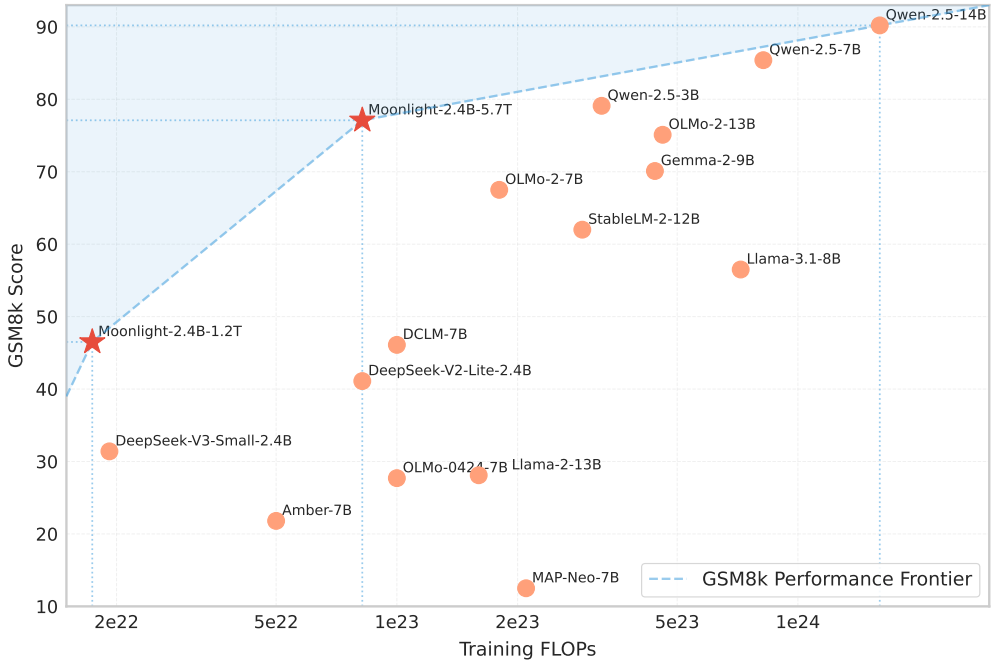


图 8: 我们采用 Muon 优化的 Moonlight 模型在 GSM8k 上的表现，与其他同类模型相比。

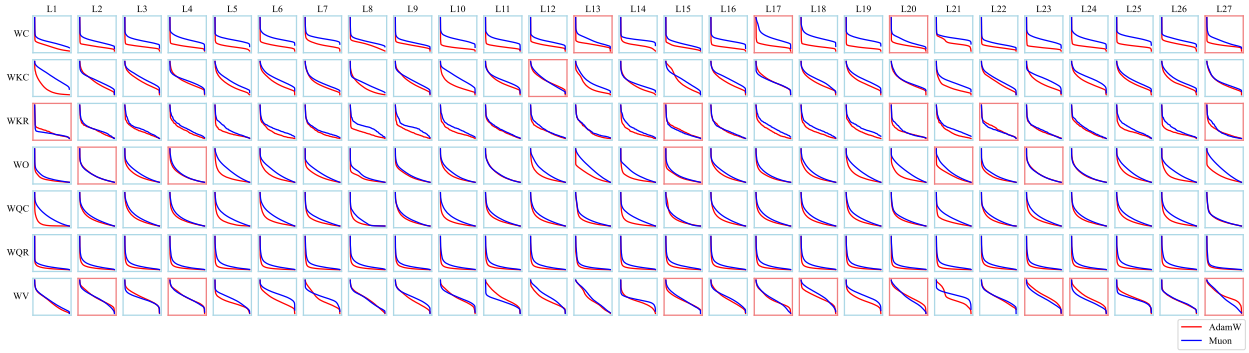


图 9: 注意力层中各权重矩阵奇异值的分布。我们用 WC 表示每层中将隐状态压缩至键与值共享潜在空间的权重矩阵，WV 表示从潜在空间上投影值的权重矩阵，WO 表示输出投影矩阵，WKR、WKC、WQR 和 WQC 分别表示带与不带 RoPE 的键和查询部分的投影矩阵。若某权重矩阵经 Muon 优化后其奇异熵低于 AdamW，则对应折线图的主干设为红色。

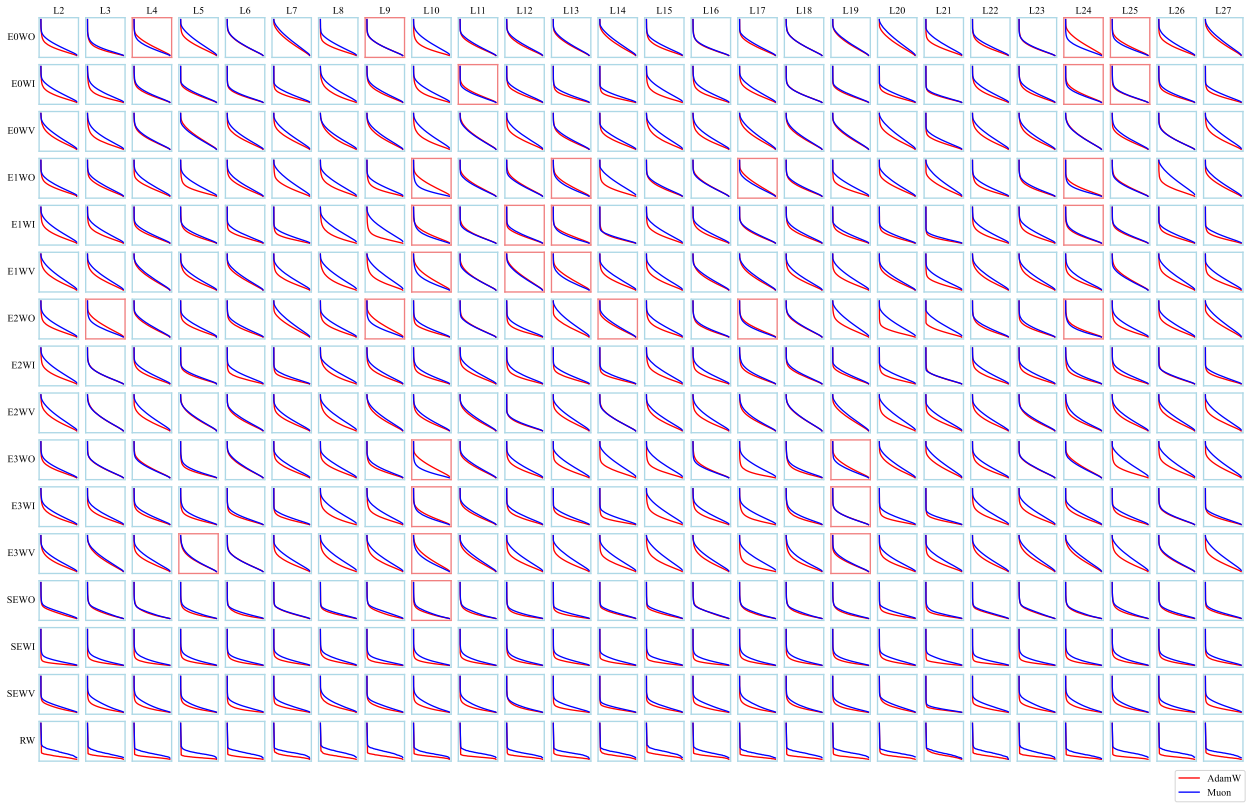


图 10: 前馈网络（FFN）层中每个权重矩阵的奇异值分布。我们使用 WI、WV 和 WO 表示涉及 SwiGLU 激活函数的 FFN 层中的权重矩阵，其中 WI 代表输入到 Swish_1 函数的投影，WV 代表与 Swish_1 激活交互的额外输入投影，WO 代表输出投影。我们使用 E0、E2、E3 表示三个任意选择的专家模型，SE 表示共享专家模型中的权重。RW 表示路由器中的权重。如果由 Muon 优化的相应权重矩阵的奇异熵低于 AdamW，我们将每个线图的脊柱设为红色。