QK-Norm 似乎损害了 Muon 优化器的大语言模型训练*

Vuk Rosić
vukrosic

今天我发现了一些奇怪的结果——QK-Norm + Muon 优化器虽然导致了更好的损失函数（loss）表现，但浪费了更多的计算资源（导致了更低秩的注意力头）。

> **设置：** 88M 参数 LLM，22 层，64 维注意力头，Muon 优化器。两次完全相同的实验——一次在查询（queries）和键（keys）上使用了 QK-RMSNorm，另一次则没有。

左图：QK-Norm 的损失函数看起来更好，但请看右侧面板，那是 **参与比（Participation Ratio, PR）**——这是一个衡量每个注意力头 64 个可用维度中实际使用了多少维度的指标（其他维度塌缩 = 变成了其他维度的倍数，不携带新信息）。

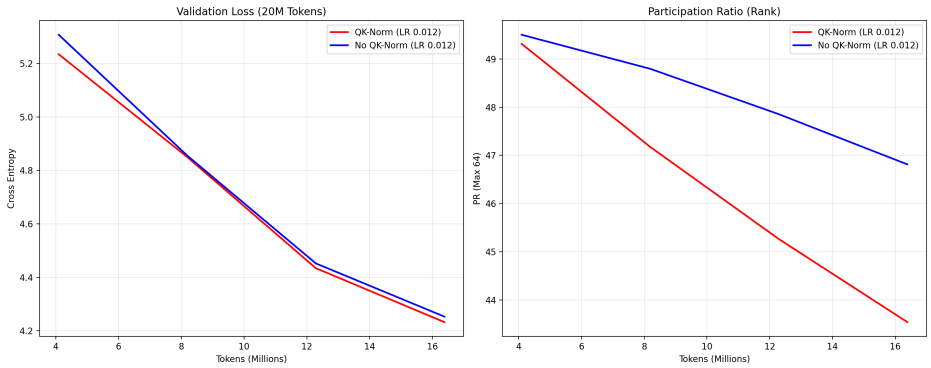

图 1: *
Loss 与秩的对比

两者的 PR 都在下降。但使用了 QK-Norm 的模型塌缩得 **更快**。到 16M token 时，QK-Norm 模型的有效维度比不使用它的版本少了约 7%。

这种差距还在不断扩大。以下是训练更久（25M token，独立实验）的情况：

*作者感谢 Novita AI 为本研究提供计算资源。

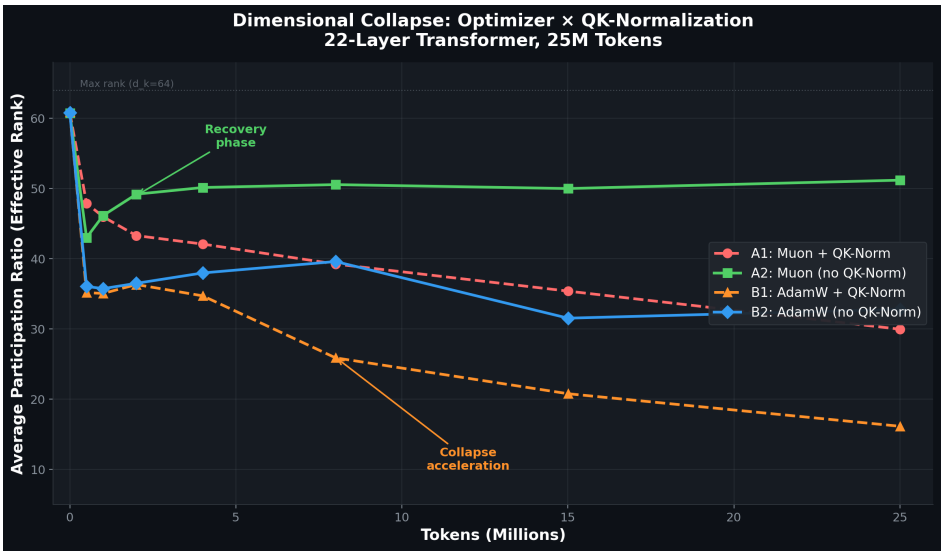

图 2: *
25M Token 轨迹

目前来看，有效维度较少的模型损失函数依然略好。但它是否构建了一些脆弱的东西?

塌缩发生在何处？

这是最有趣的地方。塌缩在不同层之间并不是均匀的。

Muon + QK-Norm (25M tokens) ——某些层基本上已经“死”了：

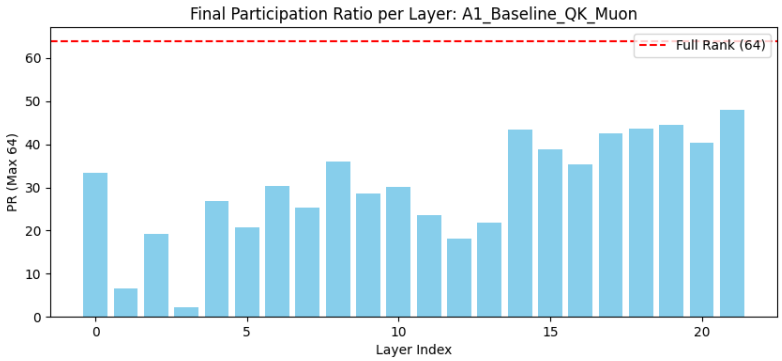

图 3: *
QK-Norm 逐层分解

看看第 1 层和第 3 层——它们的 PR 降到了 **10** 以下。在 64 个可能的维度中，这些层只使用了大约 10 个。剩下的就是我们所谓的“幽灵计算” (ghost compute) ——GPU 正在对那些

几乎没有任何贡献的维度进行数学运算。

不带 QK-Norm 的 Muon (25M tokens) ——表现均匀且具有活力：

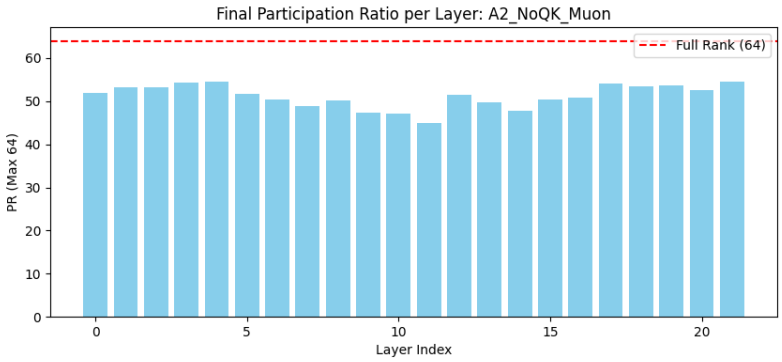

图 4: *
无 QK-Norm 逐层分解

每一层的 PR 都保持在 45 以上。没有哪一层“死掉”了。整个网络的表征带宽非常均匀。

为什么归一化（normalization）会导致某些层塌缩而其他层不会？在 QK-Norm 的运行中，第 1 层和第 3 层有什么特殊之处？（尚未解决）

悖论

以下是我们测得的数据：

衡量指标	QK-Norm	无 QK-Norm
验证集损失 (16M)	✓**4.233** (更优)	4.253
有效秩 (16M)	43.5	✓**46.8** (更高)
有效秩 (25M)	30 (正在塌缩)	✓**51** (稳定)
各层均匀性	× 存在失效层	✓所有层均活跃

结构上看起“更差”的模型实际上在预测 token 时表现略好。而内部表征更丰富的模型在损失函数上略微落后。

待解决的问题

- **结构的优势最终会转化为更好的损失函数吗？**

- **QK-Norm 是否在“作弊”？**它是否找到了一些低秩的快捷方式，虽然最小化了交叉熵，但牺牲了某些下游能力——比如上下文学习（in-context learning）、推理能力或对分布式外（OOD）数据的泛化能力?
- **Muon 的正交化压力是否已经在做 QK-Norm 所做的事情了？**
- **PR 为 51 是否真的优于 PR 为 30？**
- **在 100M+ token 时会发生什么？**

可能发生的情况是 Muon 的正交化压力与 QK-Norm 的几何约束压力之间的博弈。

在没有归一化层的情况下，Muon 可以自由地推向高秩配置。也许模型起初训练稍慢，但其内部表征保持了多样性和分布性。

这种多样性是否真的对下游能力至关重要——这是我们目前尚未回答的核心问题。