QK-Norm seems to hurt Muon optimizer LLM training*

Vuk Rosić

vukrosic

Today I found some weird results - QK-Norm + Muon optimizer leads to better loss but more wasted compute (lower rank heads)

> **Setup:** 88M parameter LLM, 22 layers, 64-dim heads, Muon optimizer. Two identical runs - one with QK-RMSNorm on queries and keys, one without.

Left: loss of QK-Norm seems better, but look at the right panel, that's the **Participation Ratio (PR)** - a measure of how many of the 64 available dimensions each attention head is actually using (others collapsed = became multiple of others, don't hold new information)

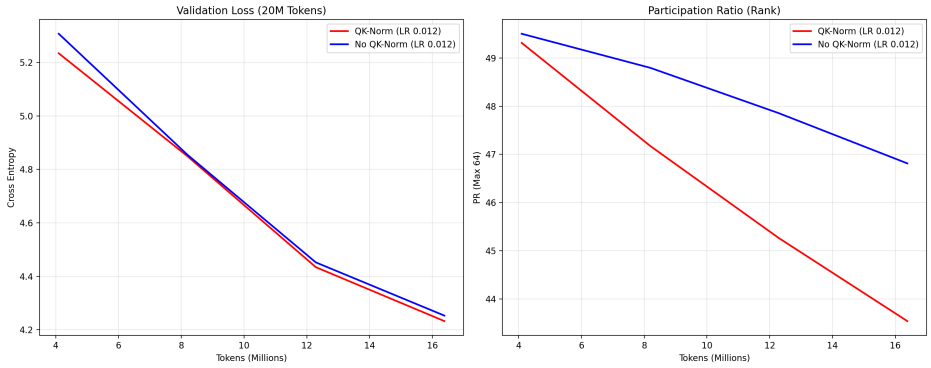

Figure 1: *
Loss and Rank Comparison

Both are dropping. But QK-Norm is collapsing **faster**. By 16M tokens, the QK-Norm model is using ∼7% fewer effective dimensions than the version without it.

This gap just keeps getting wider. Here's what happens if you train longer (25M tokens, separate experiment):

*The author would like to thank Novita AI for providing the compute resources for this study.

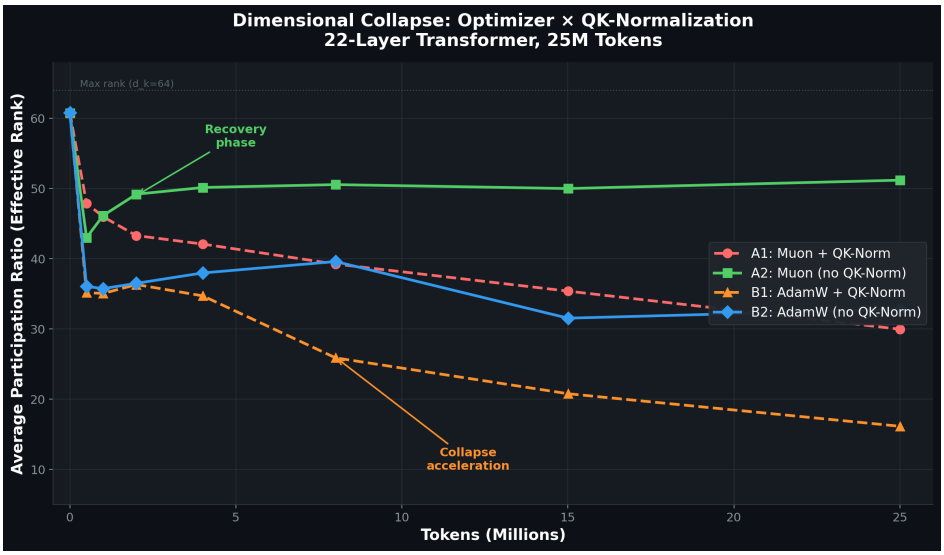

Figure 2: *
25M Trajectory

The model with fewer active dimensions still has slightly better loss... *for now*. But is it building something fragile?

Where Does the Collapse Happen?

This is where it gets interesting. The collapse isn't uniform across layers.

Muon + QK-Norm (25M tokens) - some layers are basically dead:

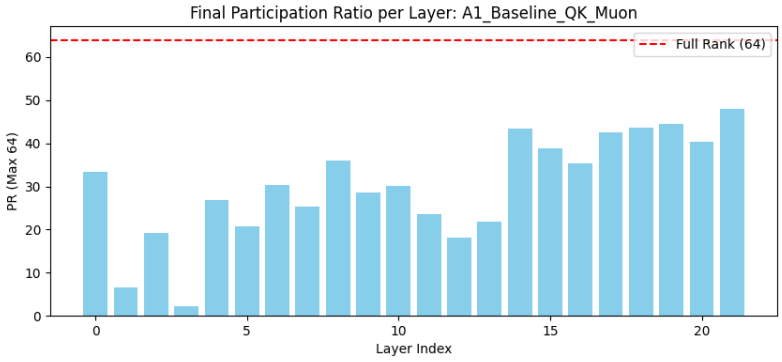

Figure 3: *
QK-Norm Layer Breakdown

Look at layers 1 and 3 - their PR dropped below **10**. Out of 64 possible dimensions, these

layers are only using ∼10. The rest is what we call "ghost compute" - the GPU is doing math on dimensions that contribute almost nothing.

Muon without QK-Norm (25M tokens) - uniform and alive:

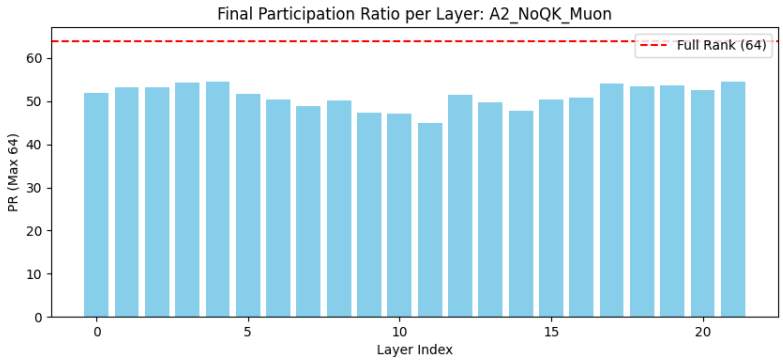

Figure 4: *
No QK-Norm Layer Breakdown

Every single layer maintains PR > 45. No layer "died." The representational bandwidth is remarkably uniform across the entire network.

Why would normalization cause some layers to collapse and not others? What's special about layers 1 and 3 in the QK-Norm run? (unanswered)

The Paradox

So here's what we're looking at:

Metric	QK-Norm	No QK-Norm
Val Loss (16M)	✓**4.233** (better)	4.253
Effective Rank (16M)	43.5	✓**46.8** (higher)
Effective Rank (25M)	30 (collapsing)	✓**51** (stable)
Layer Uniformity	× Dying layers	✓All layers alive

The model that looks "worse" structurally is actually predicting tokens slightly better. The model with richer internal representations is slightly behind on loss.

Open Questions

- **Does the structural advantage eventually translate to better loss?**

- **Is QK-Norm "cheating"?** Does it find low-rank shortcuts that minimize cross-entropy but sacrifice something downstream - like in-context learning, or reasoning, or generalization to OOD data?
- **Is Muon's orthogonalization already doing what QK-Norm does?**
- **Is a PR of 51 even better than a PR of 30?**
- **What happens at 100M+ tokens?**

What could be happening is a battle between Muon's orthogonalization pressure and QK-Norm's geometry-constraining pressure.

Without the norm layer, Muon can push into higher-rank configurations freely. Maybe the model trains a bit slower at first, but its internal representations stay diverse and distributed.

Whether that diversity actually matters for downstream capabilities - that's the real question we haven't answered yet.