When Does a Small Language Model Need QK-Norm? An Isolated Learning-Rate Characterization of Query–Key Normalization at Sub-100M Parameters

Cormac Shannon
MSc Immersive Software Engineering
University of Limerick
Limerick, Ireland
23366303

Abstract—Modern language-model architectures ship many components which have been validated using large models, however, whether the same benefits are reproduced for smaller models is actively disputed. This paper aims to produce a controlled, isolated study of one such component: Query-Key Normalisation (QK-norm) against sub 100M parameter models, defining when it becomes necessary (at which learning rate and model depths).

Index Terms—small language models, transformer architecture, training stability, query–key normalization, learning rate, ablation study

I. Introduction

The transformer architecture has quickly become a dominant choice in the natural language processing (NLP) and machine learning (ML) field. While popularly used in large scale models boasting hundreds of billions to trillions of parameters, the transformer architecture has also seen growing success in smaller models, running inference on edge-compute. These models are cheaper to run and store on low-end hardware, efficient models can run with only a few tens of millions of parameters.

Modern LLM Architecture are the culmination of many components, including but not limited to: rotary position embeddings, RMSNorm, grouped-query attention, SwiGLU, and QK-norm. Almost all of which have been validated against models billions of parameters in size [2], [5]. Although we know these techniques provided benefit in large models, it is not clearly established that these components carry their benefits to small models. Guidance for composing sub-100M parameter models is limited, and for several of the components mentioned above, the evidence from literature is contradictory regarding their benefits. Gaining clarity in this area is important, small-model pretraining is an active area of research (sample-efficient work [6], compute-constrained ablations [7]), choosing the right components matters when compute budget is tightest.

QK-Norm is a modification to a transformers self-attention step. First proposed by Henry et al. [1] for use in low-resource neural machine translation. The technique scales both the query and key vectors to a fixed magnitude before their dot product is calculated to create their attention logits. Without this step, the attention logits can grow without bound after training for many steps, causing the subsequent softmax step to create a near one-hot outcome. The learning gradients become vanishingly small, and the training loss can start spiking or diverge, requiring the learning rate to be reduced [3], [4]. Recent mechanical studies associate the reason behind logit growth to "massive-activations", wherein a small number of tokens gain extreme-magnitude activations and become the domainant influence over the key vector, forming "attention sinks" [10], [11]. QK-Norm resolves this issue by preventing unbound growth of these vectors, thus removing the failure mode.

QK-Norm was popularised by the 22-billion-parameter Vision Transformer (ViT-22B) of Dehghani et al. [2], the technique has since grown to become a near standard step in frontier language models. For example, OLMo 2 utilised RMSNorm to the Q and K vectors and found improvements to their models stability and training loss [5]. The technique has since been adopted by Gemma 2/3, Qwen3, and Chameleon.

However, QK-norm stands among several logit-control techniques: Gemma 2 soft-capped their attention logits using a smooth saturating function [12], Z-loss maintains softmax normalisers near zero preventing drifting [13], and NormFormer adds an extra normalisation layer inside each block [14]. Each of these techniques are frequently studied together, so the impact of just QK-Norm alone is difficult to observe.

Literature regarding small models is sparse, and of what exists, often entangled. Wortsman et al. provides a solid methodological template [4]. Wortsman found that instabilities found in large models can be reproduced in small models at high learning rates, QK-layernorm paired with z-loss can resolve these issues. They found this behaviour transfers across model sizes. They also introduce a learning-rate (LR) sensitivity metric that measures how far a model's final loss deviates from its optimum as the learning rate is adjusted. Recently, IMU-1 by Grigorev et al. found in 70M parameter models, the single most positively impactful architecture change was QK-Norm. This was measured via reducing the activation kurtosis [6]. However, in their study, they paired QK-Norm alongside maximal update parametrisation (μP), NorMuon, and value residuals.

It is important to note that these papers do not stand without objection. Other literature finds that some of the same ablations (RMSNorm, rotary position embeddings, grouped-query attention) do not transfer well to models sized at 70M parameters [7]. The two literatures disagree at effectively the same scale, while neither paper conducts the isolated learning-rate experiment that would settle the matter.

From this, learning rate arises as the right axis to work along. Maximal-update parametrisation details that optimal learning rate can transfer across model widths [8]. Learning rate is the hyperparameter that these stability effects are dependant on. Therefore, a faithful examination of QK-Norm requires us to study it as it relates to learning rate while maximal-update parametrisation is disabled, ensuring that its effects do not mask the behaviour we are studying.

In regards to evidence inspecting the behaviour of small models, there are two limitations surrounding them. First, evidence studying the impact on small models is confounded, Wortsman pairs qk-norm with z-loss and uses small models as a proxy for larger models [4], while IMU-1 pairs it with several other techniques (μP + NorMuon + value residuals) measuring its effectiveness through activation kurtosis at a single fixed depth and never sweeping learning rate [6]. Second, the opposing works test against a compute-optimal budget and does not isolate QK-norm or sweep learning as we desire [7]. The arising issue is that there is no study isolating the behaviour of QK-norm through different learning rates or at what model depth it becomes necessary. A second gap revolves around accessibility, most texts harness expensive GPUs, making reproducibility difficult. A laptop-reproducible study on its own is a worthwhile contribution.

This study therefore aims to isolate the effect of QK-norm and to determine at what scale a small language model benefits from it. It asks, at fewer than 100M parameters, does QK-norm — on its own — improve model convergence and raise the maximum stable learning rate? Additionally, at what learning rate does it become necessary? As a final subquestion, how does that threshold shift as the model depth is changed?

Three hypotheses can there be derived: First, the effect of QK-norm is learning-rate conditional, negligible impact or marginally harmful at low learning rates but highly impactful at higher ones (H1); that QK-Norm raises the max stable learning rate and lowers learning-rate-sensitivity (H2); and that the learning rate at which QK-norm becomes necessary lowers as depth increases (H3).

The study will investigate these hypothesis by conducting controlled ablations/changes on a nanoGPT [9] project with maximal update parametrisation disabled, AdamW set as the optimiser, while toggling QK-norm as the sole independent variable through a dense sweep of learning rates, and additionally against a small range of model depths at a model scale of roughly 10-15M parameters.

Results will be measured through the learning-rate sensitivity [4], the maximum stable learning rate, steps-to-threshold, perplexity, attention entropy and activation kurtosis.

The project will be working with small models, which has a supported precedent to produce cohesive text by using the TinyStories [15] body of text, which can train on a single-GPU as established in the Cramming study [16]. The project is designed to run on a consumer laptop (Apple M4 Pro chip), with code, seeds, and logs distributed.

References

[1] A. Henry, P. R. Dahl, et al., "Query-key normalization for transformers," in Findings of EMNLP, 2020. arXiv:2010.04245.

[2] M. Dehghani et al., "Scaling vision transformers to 22 billion parameters," 2023. arXiv:2302.05442.

[3] S. Zhai et al., "Stabilizing transformer training by preventing attention entropy collapse," in ICML, 2023. arXiv:2303.06296.

[4] M. Wortsman et al., "Small-scale proxies for large-scale transformer training instabilities," in ICLR (oral), 2024. arXiv:2309.14322.

[5] OLMo 2 Team, "2 OLMo 2 furious," 2025. arXiv:2501.00656.

[6] G. Grigorev et al., "IMU-1: Sample-efficient pre-training of small language models," 2026. arXiv:2602.02522.

[7] "Architectural trade-offs in small language models under compute constraints," 2025. arXiv:2512.20877.

[8] G. Yang et al., "Tensor programs V: Tuning large neural networks via zero-shot hyperparameter transfer," 2022. arXiv:2203.03466.

[9] A. Karpathy, "nanoGPT," GitHub repository, 2022–. https://github.com/karpathy/nanoGPT.

[10] M. Sun et al., "Massive activations in large language models," in ICML, 2024. arXiv:2402.17762.

[11] "When attention sink emerges in language models: an empirical view," 2024. arXiv:2410.10781.

[12] Gemma Team, "Gemma 2," technical report, 2024.

[13] A. Chowdhery et al., "PaLM: Scaling language modeling with pathways," 2022.

[14] S. Shleifer et al., "NormFormer: Improved transformer pretraining with extra normalization," 2021. arXiv:2110.09456.

[15] R. Eldan and Y. Li, "TinyStories: How small can language models be and still speak coherent English?," 2023. arXiv:2305.07759.

[16] J. Geiping and T. Goldstein, "Cramming: Training a language model on a single GPU in one day," in ICML, 2023. arXiv:2212.14034.