SimpleGPT

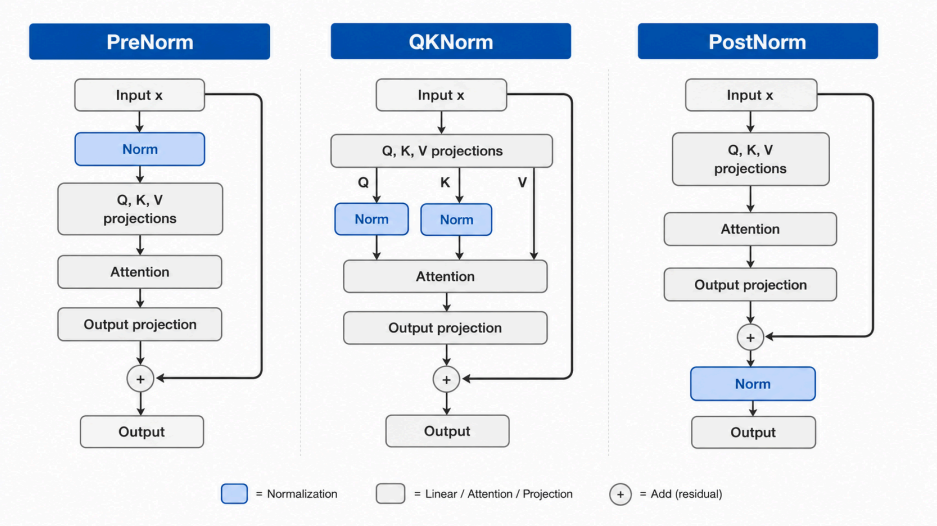
Improving GPT via A Simple Normalization Strategy

Marco Chen*, Xianbiao Qi*, Yelin He, Jiaquan Ye, Rong Xiao

ICML 2026

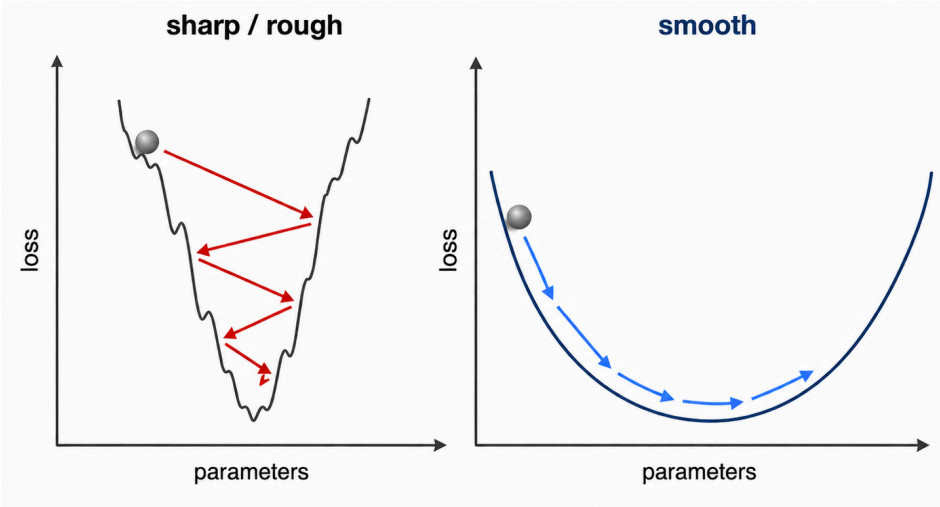
Problem: where should normalization go?

Normalization stabilizes Transformer training



Where we place normalization is often decided **heuristically**

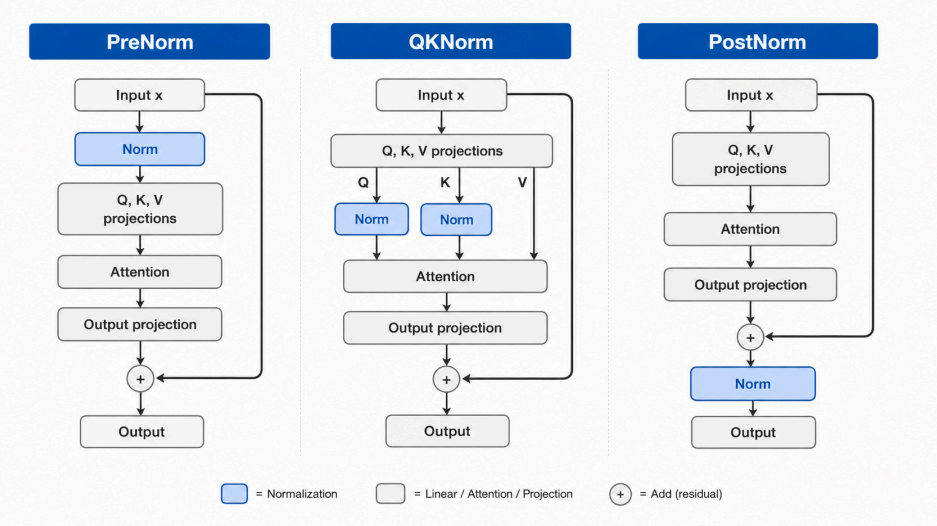
Perspective: normalization and optimization geometry



smoother loss landscape ⇒ less likely to overshoot or oscillate.

Problem: where should normalization go?

Normalization stabilizes Transformer training



Where we place normalization is often decided **heuristically**

Perspective: normalization and optimization geometry

$$\eta \leq \frac{2}{\beta}, \quad \beta \approx \sup_{\boldsymbol{x}} \|\boldsymbol{H}_{\boldsymbol{x}\boldsymbol{x}}\|_2$$

The maximum stable LR is controlled by the curvature of the loss landscape.

Question: Can we design a simple architectural rule that reduces curvature and makes training more stable?

SimpleNorm: a unified linear and normalization operation

In SimpleNorm, linear projection is followed by normalization

Given activation $\boldsymbol{x} \in \mathbb{R}^{n \times d}$ and weight matrix $\boldsymbol{W} \in \mathbb{R}^{m \times n}$, we define SimpleNorm as:

$$\Psi(\boldsymbol{x}) = \mathrm{Norm}(\boldsymbol{W}\boldsymbol{x})$$

Instantiate $\mathrm{Norm}(\cdot)$ with RMSNorm

$$\Psi(\boldsymbol{x}\ ;\ \boldsymbol{W}, \boldsymbol{\gamma}) = \boldsymbol{\gamma} \odot \sqrt{d} \frac{\boldsymbol{W}\boldsymbol{x}}{\|\boldsymbol{W}\boldsymbol{x}\|_2}$$

Projection and normalization are now one **unified** operator

SimpleNorm stabilizes activation scale by construction

$$\|\Psi(\boldsymbol{x})\|_2 = \Theta\left(\sqrt{d}\right)$$

normalization prevents activation scale from drifting with **depth** or **weight** growth

Short Justification

$$\Psi(\boldsymbol{x}) = \boldsymbol{\gamma} \odot \sqrt{d}\frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_2} \Longrightarrow \boldsymbol{\gamma}_{\min}\sqrt{d} \leq \|\Psi(\boldsymbol{x})\|_2 \leq \boldsymbol{\gamma}_{\max}\sqrt{d}$$

Hence, $\|\Psi(\boldsymbol{x})\|_2 = \Theta\left(\sqrt{d}\right)$ provided $\boldsymbol{\gamma} = \Theta(1)$, which holds in typical settings.

SimpleNorm: Mechanism II - Smoother Local Curvature

Plain Linear Projection

$$\left\| \boldsymbol{H}_{\boldsymbol{xx}}^{\mathrm{lin}} \right\|_2 = \Theta\left(\|\boldsymbol{W}\|_2^2 \left\| \boldsymbol{H}_{\boldsymbol{yy}} \right\|_2 \right)$$

Curvature grows with the spectral norm of the weight matrix.

SimpleNorm Projection

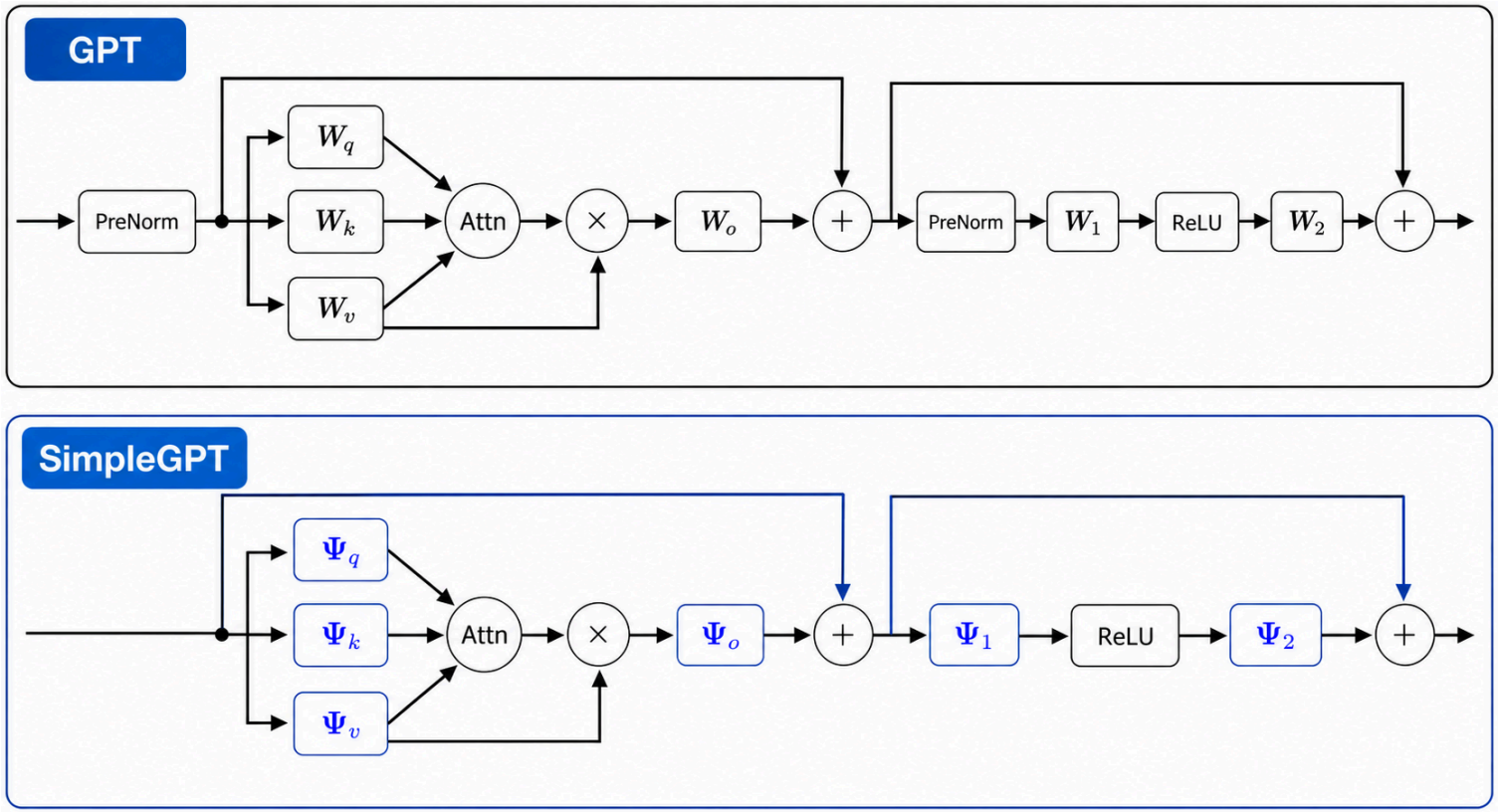
$$\left\| \boldsymbol{H}_{\boldsymbol{xx}}^{\mathrm{sn}} \right\|_2 = \Theta\left(\left\| \boldsymbol{H}_{\boldsymbol{yy}} \right\|_2 \right)$$

Immediate normalization removes this weight-scale dependence.

The weight norm $\|\boldsymbol{W}\|_2^2$ generally grows during training. By removing the weight-scale dependence in its Hessian, SimpleNorm leads to a **smoother local loss landscape**.

SimpleGPT Architecture

Standard GPT-style models choose among PreNorm, QKNorm, PostNorm, and related placements. **SimpleGPT** replaces **every linear map** with **SimpleNorm**.



Experiments: larger tolerable learning rates

Experimental Setup

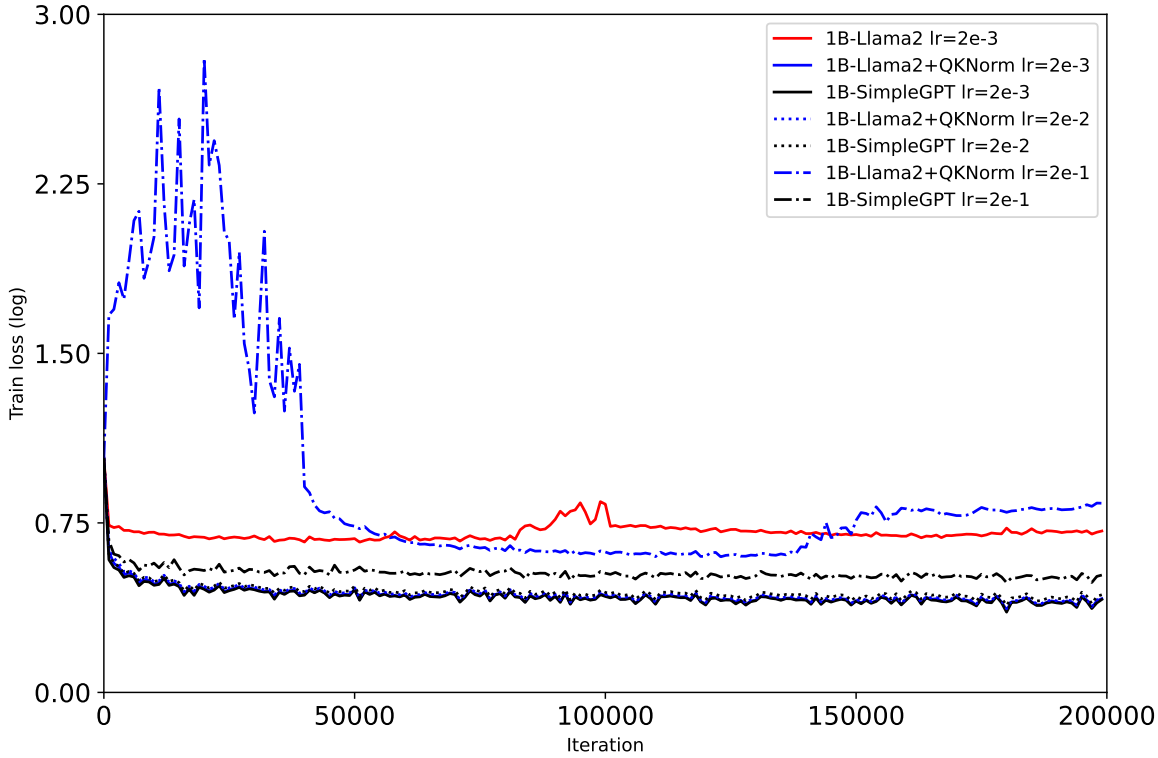
`nanoGPT`, `Llama2`, and `Llama3` across 1B to 8B parameters using C4 dataset.

Largest Tolerable LR

PreNorm: unstable at relatively small LR.

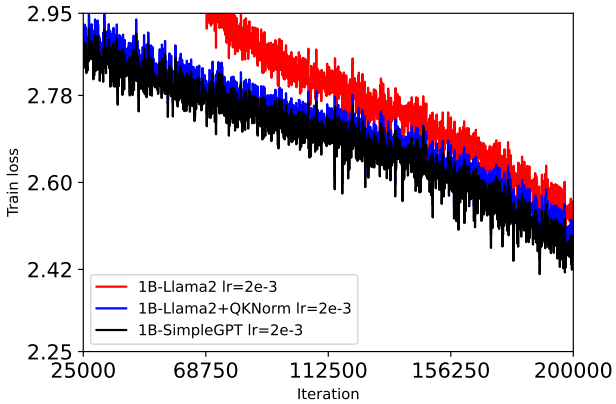
QKNorm: improves stability, then breaks as LR increases.

SimpleGPT: remains stable at much larger LR.



SimpleGPT improves training loss from 1B to 8B

Llama2-1B



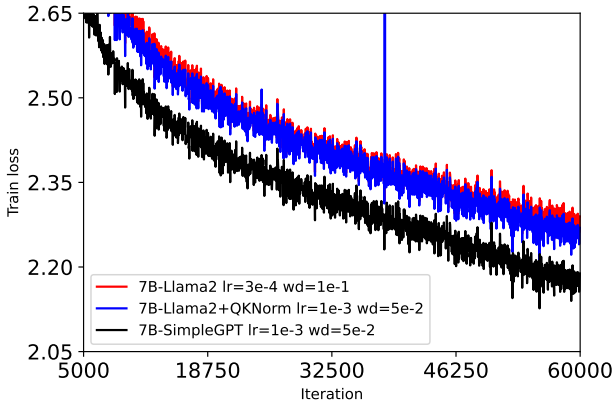
1B-Llama2 lr=2e-3: last = 2.520

1B-Llama2+QKNorm lr=2e-3: last = 2.478

1B-SimpleGPT lr=2e-3: last = 2.446

SimpleGPT improves over PreNorm and QKNorm.

Llama2-7B



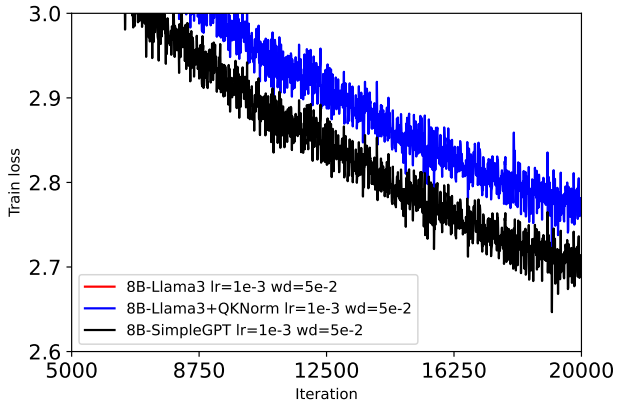
7B-Llama2 lr=3e-4 wd=1e-1: last = 2.311

7B-Llama2+QKNorm lr=1e-3 wd=5e-2: last = 2.290

7B-SimpleGPT lr=1e-3 wd=5e-2: last = 2.208

At 7B scale, SimpleGPT's improvement gap widens.

Llama3-8B



8B-Llama3 lr=1e-3 wd=5e-2: last = 4.640

8B-Llama3+QKNorm lr=1e-3 wd=5e-2: last = 2.752

8B-SimpleGPT lr=1e-3 wd=5e-2: last = 2.678

Improvement persists from Llama2 $\rightarrow$ Llama3.

Downstream tasks: gains transfer beyond training loss

Held-out Evaluation

We measure bpb on held-out validation, lower is better

Metric	**SimpleGPT**	QKNorm
C4 val bpb	**0.7999**	0.8327
Wiki bpb	**0.7896**	0.8188

Downstream benchmarks

Accuracy, higher is better

Benchmark	**SimpleGPT**	QKNorm
WinoGrande	**53.99**	52.88
PIQA	65.61	**65.77**
HellaSwag	**45.92**	45.33

SimpleGPT improves C4 and Wiki bpb, and remains competitive or better on downstream accuracy.

SimpleNorm is a simple architectural rule that unifies linear projection and normalization

$$\Psi(\boldsymbol{x}) = \mathrm{Norm}(\boldsymbol{W}\boldsymbol{x})$$

SimpleNorm **stabilizes activation scale** and **smooths local curvature**.

By systematically applying SimpleNorm, SimpleGPT enables **larger stable learning rates** and **improves GPT-style models across architectures and scales**.

Limitations and Future Work

- Extra normalization overhead, around 3% training-time increase.
- More validation needed for MoE, multimodal, and larger-scale production training.