NORMALISATION CHOICES IN DECODER-ONLY TRANSFORMERS ARE REGIME-DEPENDENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Three claims about normalisation in Transformers circulate as folk theorems: Pre-LN is stable and Post-LN is not; RMSNorm is a free drop-in for LayerNorm; and QK-Norm stabilises attention. They are rarely tested against each other in one codebase, on one corpus, with shared seeds. We do so with 175 runs of a GPT-2-style decoder (21.3 M non-embedding parameters at depth 12) on FineWeb-Edu, comparing six variants arranged as two single-axis chains rooted at Pre-LN and Post-LN, five of them across four depths, a learning-rate sweep, reruns at the selected rates, a gradient-clip stress test and per-layer gradient diagnostics. The apparent answer to "which variant is better?" changes materially with the optimisation regime. At a shared learning rate the Post-LN penalty is $+0.0495$ nats; at each variant's sweep-selected learning rate it is $+0.1442$ nats, a 2.9-fold change from a single hyperparameter. A single-architecture chain attributes the Post-LN rescue cleanly: swapping LayerNorm for RMSNorm moves the loss by $+0.0001$ nats, while adding QK-Norm moves it by -0.0504 nats and lifts the Post-LN learning-rate ceiling threefold. That headroom is depth-limited: at depth 48 five variants converge to within 0.021 nats, but at 10^{-3} both Post-LN variants—QK-Norm included—collapse whether gradient clipping is relaxed $100\times$ or left at its default, so of the two variables crossed the learning rate determines the outcome and the clip does not. The failures are not blow-ups: collapsed runs settle $+0.020$ nats from the corpus unigram entropy, which we measure directly. We release all 175 per-run artefacts with a pipeline that regenerates every number here.

1 INTRODUCTION

Practitioners carry three folk theorems about normalisation in Transformers: that Pre-LN is stable and Post-LN is not, especially at depth (Xiong et al., 2020; Liu et al., 2020); that RMSNorm is a faster drop-in for LayerNorm with no loss in quality (Zhang & Sennrich, 2019; Touvron et al., 2023); and that QK-Norm—normalising queries and keys before the dot product—stabilises attention logits (Henry et al., 2020; Dehghani et al., 2023).

Each claim has good evidence behind it, and each has been studied against a moving learning rate: Wortsman et al. (2024) sweep the learning rate for QK-LayerNorm under Pre-LN, Rybakov et al. (2024) do so for several attention-normalisation variants, and Kim et al. (2025) sweep learning rates and seeds across Pre-LN, Post-LN and Peri-LN. What is still missing is a design that separates the three axes from *each other*. Where QK-Norm has been placed outside the Pre-LN layout it has arrived bundled with other changes—OLMo 2 adopts reordered normalisation and QK-Norm together (OLMo Team, 2025), HybridNorm combines QKV-normalisation with a Post-Norm feed-forward block (Zhuo et al., 2025)—so the contribution of the normalisation *function* cannot be told apart from that of QK-Norm itself. This paper asks what the three folk theorems look like when each axis is moved on its own, and when the comparison is controlled along the dimensions a normalisation change actually perturbs: learning rate, training horizon, depth, and gradient-clipping policy.

We train a GPT-2-style decoder-only Transformer (Radford et al., 2019) of about 21 M non-embedding parameters on FineWeb-Edu (Penedo et al., 2024) and compare six normalisation variants arranged as two parallel single-axis chains, PreLN $\rightarrow$ RMSNorm-PreLN $\rightarrow$ QKNorm-PreLN and PostLN $\rightarrow$ RMSNorm-PostLN $\rightarrow$ QKNorm-PostLN, so that every headline comparison is be-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tween rows differing on one axis, with a DeepNet (Wang et al., 2024) baseline as an alternative rescue. The study spans 175 runs, 64.0 B training tokens and 311 GPU-hours in thirteen blocks (Table 1).

Findings. The through-line is that *normalisation comparisons are regime-dependent*:

- **A single hyperparameter changes the headline number 2.9-fold.** At a shared learning rate of 3×10^{-4} the Post-LN penalty relative to Pre-LN at depth 12 is $+0.0495$ nats (paired range 0.0447–0.0535). Re-running the same architectures at each variant's sweep-selected learning rate widens it to $+0.1442$ nats (0.1413–0.1472), because Pre-LN tolerates 3×10^{-3} without collapse while Post-LN cannot leave 3×10^{-4}.
- **QK-Norm, not RMSNorm, is the load-bearing axis of the Post-LN rescue.** Within a single architecture, LayerNorm $\rightarrow$ RMSNorm on the Post-LN side moves the loss by $+0.0001$ nats with a paired spread of ±0.001; adding QK-Norm moves it by -0.0504 nats (0.0452–0.0531, Cohen's $d = 11.2$) and lifts the Post-LN learning-rate ceiling from 3×10^{-4} to 10^{-3}. Under selected learning rates that combination reaches 3.7728 nats, the lowest loss we record at that depth: in our controlled chain essentially all of the measured rescue is attributable to QK-Norm rather than to the LayerNorm $\rightarrow$ RMSNorm substitution it is often bundled with.
- **QK-Norm's learning-rate headroom does not survive to depth 48.** With 20 k steps and a conservative learning rate, the five variants run at depth 48 reach parity to within 0.021 nats. Raise the learning rate to 10^{-3} and both Post-LN variants—QK-Norm included—collapse, while Pre-LN survives every setting it was run at. Crossing the clip with the learning rate isolates which variable is responsible: at 3×10^{-4} nothing collapses at either clip setting, and at 10^{-3} everything Post-LN collapses at either. The failures are not blow-ups—all but one of the failing runs settle within 250 steps at $+0.020$ nats from the measured unigram entropy of the corpus, and none produces a non-finite loss, so a pipeline that only checks for NaNs would score every one as a success.

Scope. These are statements about a small ($\approx$ 21 M non-embedding), GPT-2-style, FineWeb-Edu regime at depths up to 48, with AdamW and no μP; against the 0.4–3 B-parameter studies cited above they are a controlled small-scale probe, not a frontier-scale claim. Our contribution is not a new normalisation scheme but a controlled measurement of existing ones—single-axis chains on both layouts, exact seed pairing, and the gradient clip treated as an experimental variable—reported with the statistics three seeds can support.

2 Related work

Placement and normalisation function. The original Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2019) apply LayerNorm (Ba et al., 2016) *after* the residual addition. Xiong et al. (2020) showed that Post-LN produces large gradients at initialisation, which is why it needs warm-up, while Pre-LN does not; GPT-2 (Radford et al., 2019) and its descendants use Pre-LN, as does machine translation (Nguyen & Salazar, 2019). Remedies for Post-LN include adaptive initialisation (Liu et al., 2020), DeepNet's α/β scaling (Wang et al., 2024), and extra in-block normalisation (Shleifer et al., 2021). A recent line revisits placement at scale: Li et al. (2025) apply Post-LN early and Pre-LN deep, arguing Pre-LN under-trains deep layers—developed further by Sun et al. (2025); Kim et al. (2025) compare Pre-LN, Post-LN and Peri-LN at 400 M–3.2 B parameters while sweeping learning rates and seeds; Zhuo et al. (2025) combine QKV-normalisation with a Post-Norm feed-forward block; Chen & Wei (2026) report stable Post-LN training past $1{,}000$ layers once the residual is replaced by a highway-style connection. Closest to our depth-48 failures, Wang et al. (2026) derive Post-Norm collapse from attention amplification and a loss of gradient repair, identifying its endpoint as the token-frequency distribution with vanishing gradients—which our runs recover independently and locate against the measured corpus entropy (Section 4.3). These establish that placement interacts strongly with optimisation at scale, but do not separate the normalisation function from attention-logit normalisation within a fixed layout, nor test whether that separation survives a change of learning rate, depth or clipping policy, which is the gap the single-axis chains below fill. RMSNorm (Zhang & Sennrich, 2019) drops LayerNorm's mean-centering

and bias and is widely adopted in Pre-LN decoder models (Touvron et al., 2023; OLMo Team, 2025; Gemma Team, 2025); the quality-neutrality claim is generally quoted from that setting, and we test it on both sides of the layout axis.

Attention-logit control and clipping. Henry et al. (2020) proposed normalising queries and keys before the dot product. Dehghani et al. (2023) used LayerNorm on Q and K to remove a divergence caused by growing attention logits at 22 B parameters; Wortsman et al. (2024) showed at small scale that QK-LayerNorm reduces learning-rate sensitivity; Zhai et al. (2023) traced a related failure to attention-entropy collapse; Rybakov et al. (2024) find at 830 M parameters that extending normalisation to the QKV output, or pairing QK-LayerNorm with softmax capping, permits a $1.5\times$ higher learning rate than QK-LayerNorm alone. QK-Norm now appears in Chameleon (Chameleon Team, 2024), OLMo 2 (OLMo Team, 2025) and Gemma 3 (Gemma Team, 2025), which replaced Gemma 2's logit soft-capping (Gemma Team, 2024) with it. Our contribution relative to Wortsman et al. (2024) and Rybakov et al. (2024), both of which keep a Pre-LN backbone, is to place QK-Norm on the Post-LN side as well, isolate it from a concurrent LayerNorm $\rightarrow$ RMSNorm change with a single-axis control—a separation the OLMo 2 and HybridNorm recipes leave bundled—and test whether its learning-rate tolerance survives when gradient clipping is relaxed. Clipping (Pascanu et al., 2013) is the standard safety rail; Takase et al. (2023) argue that keeping the gradient norm small requires small sub-layers and a large shortcut. Our stress test treats the clip threshold as an experimental variable rather than a fixed part of the recipe.

3 Experimental design

3.1 Model, data and optimisation

We use a GPT-2-style decoder (Radford et al., 2019) implemented directly in PyTorch (Paszke et al., 2019) with no trainer framework, with an output head tied to the token embedding (Press & Wolf, 2017; Inan et al., 2017). Width and head count are fixed throughout (`emb_dim` $= 384$, 6 heads of dimension 64, FFN multiplier 4, context 512, vocabulary 50,257), so parameter count is matched across variants within each depth to within 0.05% (Appendix A). The two residual layouts are Post-LN (Vaswani et al., 2017) and Pre-LN (Xiong et al., 2020; Radford et al., 2019):

$$\text{Pre-LN:}\quad x' = x + \mathrm{Attn}(\mathrm{Norm}(x)), \qquad x'' = x' + \mathrm{FFN}(\mathrm{Norm}(x')), \tag{1}$$

$$\text{Post-LN:}\quad x' = \mathrm{Norm}(x + \mathrm{Attn}(x)), \qquad x'' = \mathrm{Norm}(x' + \mathrm{FFN}(x')), \tag{2}$$

where Norm is either LayerNorm (Ba et al., 2016) or RMSNorm (Zhang & Sennrich, 2019), with $\epsilon = 10^{-5}$. Our QK-Norm applies a per-head RMSNorm, with separate learned scale vectors for queries and keys, along the head dimension before the dot product, retaining the usual $1/\sqrt{d_h}$ factor:

$$Q' = \mathrm{RMSNorm}_q(Q), \qquad K' = \mathrm{RMSNorm}_k(K), \qquad \text{scores} = Q'K'^{\top}/\sqrt{d_h}. \tag{3}$$

This is the RMSNorm form of QK-LayerNorm (Dehghani et al., 2023; Wortsman et al., 2024), not the ℓ_2 QKNorm of Henry et al. (2020); "QK-Norm" here always means the per-head RMSNorm variant. Table 5 (Appendix A) lists the variants. DeepNet (Wang et al., 2024) uses the Post-LN layout with the residual scaled by $\alpha = (2N)^{1/4}$ and Xavier-normal initialisation (Glorot & Bengio, 2010) with gain $\beta = (8N)^{-1/4}$ on selected projections, so it differs from PostLN on the initialisation scale of *every* linear layer (Appendix F).

All runs train on the `sample-10BT` subset of FineWeb-Edu (Penedo et al., 2024), streamed and tokenised with the GPT-2 byte-pair encoder (Radford et al., 2019; Sennrich et al., 2016) into non-overlapping 512-token windows. The first 10,000 documents are held out as a fixed, pre-tokenised validation set (10.3 M tokens; 20,140 windows); training streams from the 10,000th onward through a seeded reservoir shuffle, and we verified the streaming library refuses to reorder shards underneath a `skip`, so the held-out prefix stays held out for every seed. Validation loss is mean cross-entropy over all token positions, in nats, on a random sample of 25 batches of 64 windows ($\approx 8\%$ of the set), reported at the *last* evaluation. Every run uses AdamW (Loshchilov & Hutter, 2019) with $\beta = (0.9, 0.95)$, weight decay 0.1 on matrices, linear warm-up then cosine decay (Loshchilov & Hutter, 2017) to 10% of the peak, batches of 64×512 tokens, and global-norm gradient clipping at 1.0 unless stated. A 20 k-step run sees 655 M tokens; the gradient norm is recorded *before* clipping (Appendix A).

Table 1: Study blocks. LR $= 3 \times 10^{-4}$ and clip $= 1.0$ unless stated. "Closure" is the Post-LN final-norm convention the block was trained with (Section 3.3); it is a no-op for Pre-LN variants.

Block	Purpose	Design	Closure	Runs
A	Matched LR	5 variants $\times$ 3 seeds $\times$ 20k, $d = 12$	`ln_f`	15
B	Depth scan	5 variants $\times$ $d \in \{6, 12, 24, 48\}$ $\times$ 2 seeds $\times$ 5k	`ln_f`	40
1	LR sweep	5 variants $\times$ LR $\in \{10^{-4}, 3\times10^{-4}, 10^{-3}, 3\times10^{-3}\}$ $\times$ 1 seed $\times$ 5k, $d = 12$	`ln_f`	20
2	Selected LR	5 variants $\times$ 3 seeds $\times$ 20k, $d = 12$	`ln_f`	15
3	Stress test	{PreLN, PostLN, QKNorm-PostLN} $\times$ 3 LRs $\times$ 3 seeds $\times$ 12k, $d = 48$, clip $= 100$	`ln_f`	27
4	Converged $d = 48$	5 variants $\times$ 2 seeds $\times$ 20k, $d = 48$	`ln_f`	10
5	DeepNet	DeepNet-PostLN $\times$ $d \in \{12, 24, 48\}$ $\times$ 3 seeds $\times$ 20k	`ln_f`	9
6	RMSNorm-PostLN control	4 LRs $\times$ 1 seed $\times$ 5k; 3 seeds $\times$ 20k at 3×10^{-4}; $d = 12$	Identity	7
7	Closure calibration	{PostLN, QKNorm-PostLN} $\times$ 3 seeds $\times$ 20k, $d = 12$	Identity	6
8	Clip isolation	{PostLN, QKNorm-PostLN} $\times$ 3 seeds $\times$ 1k, $d = 48$, LR $= 10^{-3}$, clip $= 1.0$	`ln_f`	6
9	Canonical closure, selected LR	{QKNorm-PostLN at 10^{-3}, PreLN at 3×10^{-3}} $\times$ 3 seeds $\times$ 20k, $d = 12$	Identity	6
10	Canonical closure, $d = 48$	Block 8 with the closure flipped; all else identical	Identity	6
11	Determinism check	{PreLN, PostLN} $\times$ 2 replicates $\times$ 2k, $d = 12$, run in two separate launches	Identity	8
	Total			175

3.2 Study blocks

Table 1 lists the thirteen blocks. Block A is the precise comparison at fixed depth and learning rate; Block 1 sweeps the learning rate per variant and Block 2 re-runs Block A at the selected rates. Block B scans depth at a short horizon and Block 4 supplies converged depth-48 cells. Block 3 is the stress test with the clip relaxed $100\times$ and Block 8 the cell that separates the clip from the learning rate. Block 5 is the DeepNet baseline; Block 6 adds the RMSNorm-PostLN control that makes the Post-LN chain single-axis; Block 7 is the closure calibration described next; Blocks 9 and 10 re-measure that calibration in the two regimes it did not reach; Block 11 is the determinism check.

3.3 Closure, seed pairing and statistical protocol

Two Post-LN closures. Blocks A, B and 1–5 were trained before a code change affecting only the Post-LN family: the model then applied its final pre-output norm (`ln_f`) to *every* layout, including Post-LN, where the block already ends in a norm. That is non-canonical and was corrected to `Identity` before Blocks 6 and 7. Block 7 re-trains the two Post-LN cells of Block A canonically, seed for seed; isolating the closure from the block effect gives -0.0001 nats (95% CI ± 0.0003), a near-no-op as $\mathrm{LN}(\mathrm{LN}(x)) \approx \mathrm{LN}(x)$ predicts. Block 7's runs are canonical end-points, so with Block 6 they form a Post-LN chain whose three rows share one architecture; that chain is what Section 4.2 reports. Appendix B gives per-seed values and states which downstream claims the calibration covers directly.

Pairing is exact within a block. Model construction consumes the same random-number draws for every variant—verified by hashing the global RNG state after constructing each one—and the training-data shuffle and validation sampling are driven from that same state, so two runs of a block sharing a seed see the same training-token order *and* the same validation batches. Detrended residuals of late evaluations correlate at $r = 0.995$ across the variants of a block at a shared seed (minimum 0.989). Validation-sampling noise (≈ 0.009 nats per evaluation) inflates the per-variant $\pm$ columns but cancels in every paired difference, which is why paired spreads of 0.001–0.016 nats sit well inside per-variant standard deviations of 0.006–0.021.

Pairing does not survive a change in the data stream. The same configuration and seed run in Blocks A and 2 draws different validation batches ($r \approx 0$) and lands up to 0.018 nats away (s.d. 0.016). That is not GPU non-determinism: Block 11 shows training to be bit-reproducible on this hardware, with all 8 comparisons of identical configurations agreeing to the last representable digit at every evaluation. What differed is therefore an input to the block, and since every run in a block shares its inputs, the difference is a common-mode offset rather than independent noise: we treat

Table 2: Depth 12, 20 k steps, three seeds: Block A (+ Block 6†) at a shared learning rate, and Block 2 at each variant's selected rate. ± is across-seed sample s.d. and includes ≈ 0.009 nats of validation-sampling noise that the paired columns of Table 10 do not. †RMSNorm-PostLN comes from a different block, so its comparison with the others carries the cross-block offset of Section 3.3; its selected LR equals its matched one, so Block 6 already *is* its selected-LR result. Δ columns are computed from unrounded values.

	Matched LR 3×10^{-4}		Per-variant selected LR		
Variant	Final val loss	Δ vs PreLN	LR	Final val loss	Δ vs PreLN
PreLN	3.9426 ± 0.0070	–	3×10^{-3}	3.8482 ± 0.0211	–
RMSNorm-PreLN	3.9434 ± 0.0057	+0.0008	3×10^{-3}	3.8458 ± 0.0204	-0.0025
QKNorm-PreLN	$\mathbf{3.9141 \pm 0.0073}$	-0.0284	1×10^{-3}	3.7880 ± 0.0153	-0.0602
PostLN	3.9921 ± 0.0106	+0.0495	3×10^{-4}	3.9924 ± 0.0183	+0.1442
RMSNorm-PostLN†	3.9924 ± 0.0171	+0.0499	3×10^{-4}	*as matched*	–
QKNorm-PostLN	3.9413 ± 0.0176	-0.0013	1×10^{-3}	$\mathbf{3.7728 \pm 0.0125}$	-0.0755

0.018 nats as a bound on cross-block comparisons, not a standard error. Every paired difference in this paper is taken inside one block or across a pair whose pairing we have verified. Appendix C gives the measurements and the candidate cause.

What we report. For every comparison we give the mean paired difference $\bar{\Delta}$, its exact range across seeds, Cohen's d on the paired differences (Cohen, 1988), the paired t-test p, and the Bonferroni-corrected p over the 10 genuinely paired comparisons in the block; we lead with $\bar{\Delta}$ and its range. Comparisons that cross a block are not paired in fact: we give their difference and range but no d or p, and they do not enter the family. 1 matched-block pair and the four Post-LN-penalty pairs of the selected-LR block survive Bonferroni at $\alpha = 0.01$ (4 of 10 and 8 of 10 at $\alpha = 0.05$). With three paired observations only very large effects clear family-wise correction—$d \gtrsim 18$ at $\alpha = 0.01$ over ten pairs—so d of 6–11 should be read as "the sign is the same in every seed and the effect is many times the paired noise", not as a failed test.

4 Results

4.1 Depth 12: the learning rate moves the headline number

Table 2 gives Block A (matched LR, with the Block 6 control) beside Block 2 (each variant at its selected rate from Block 1), both at depth 12, 20 k steps, three seeds. Figure 1 gives the sweep those best values come from. Paired ranges, Cohen's d and p-values for every comparison are in Appendix D (Tables 10 and 11); the sweep grid itself is Table 12 there.

At a shared learning rate. QK-Norm helps: QKNorm-PreLN beats PreLN by 0.0284 nats with the same sign in every seed ($d = 6.7$). Post-LN trails Pre-LN by 0.0495 (0.0447–0.0535, $d = 11.1$), consistent in direction with Xiong et al. (2020). RMSNorm matches LayerNorm on the Pre-LN side—the paired difference is +0.0008 nats and changes sign across seeds, which is what "no detectable effect" looks like—while throughput rises 3.2%. And the combined variant closes the Post-LN gap entirely at this rate (−0.0013 nats vs PreLN), an effect Section 4.2 attributes almost entirely to one of its two axes.

The sweep. The Pre-LN family is robust across the whole grid, best at the top end. Vanilla PostLN cannot tolerate LR $\geq 10^{-3}$ at depth 12: its best value is 3×10^{-4}, the very rate used in Block A. RMSNorm does not move that ceiling—RMSNorm-PostLN and PostLN are indistinguishable at every grid point and collapse at the same two—whereas QK-Norm moves it 3×: QKNorm-PostLN survives 10^{-3} and there records the sweep's lowest loss (4.120 nats, 0.48 better than its next-best setting; Appendix D).

At selected learning rates. The rerun makes three things visible. **First, the combined variant wins**: QKNorm-PostLN reaches 3.7728 nats, beating PreLN by 0.0755 (0.0670–0.0848, $d = 8.5$)—about 60× the matched-LR margin, which was inside the noise. **Second, the Post-LN penalty**

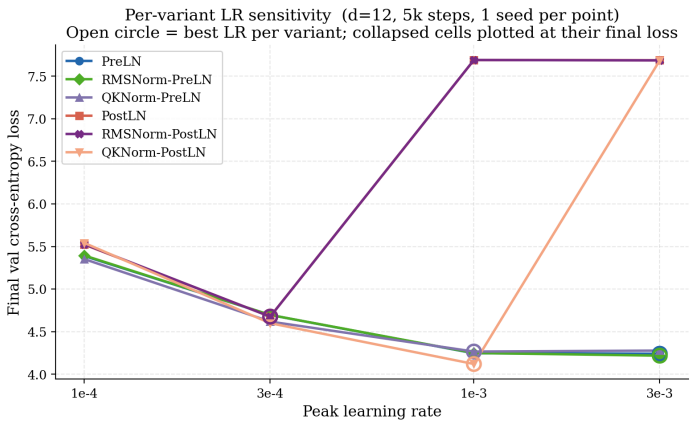


Figure 1: Learning-rate sensitivity per variant (Block 1 + Block 6), depth 12, 5 k steps, one seed per point. Open circles mark each variant's selected rate, which is its rate in Block 2. The Post-LN family falls off a cliff once its ceiling is crossed; the Pre-LN family is nearly flat from 10^{-3} to 3×10^{-3}.

Table 3: The canonical single-axis Post-LN chain: PostLN and QKNorm-PostLN from Block 7, RMSNorm-PostLN from Block 6; depth 12, LR $= 3 \times 10^{-4}$, 20 k steps, the same three seeds throughout. $\bar{\Delta}$ is the change along the arrow.

Step	Axis changed	Mean loss	$\bar{\Delta}$	Δ range	Cohen's d	p
PostLN → RMSNorm-PostLN	LayerNorm → RMSNorm	3.9923 → 3.9924	**+0.0001**	[-0.0012, +0.0009]	0.1	0.90
RMSNorm-PostLN → QKNorm-PostLN	add QK-Norm	3.9924 → 3.9420	**-0.0504**	[-0.0531, -0.0452]	11.2	0.003
PostLN → QKNorm-PostLN	both (full rescue)	3.9923 → 3.9420	**-0.0503**	[-0.0543, -0.0446]	9.8	0.003

grows: PostLN, stuck at 3×10^{-4}, now trails PreLN at 3×10^{-3} by 0.1442 nats (0.1413–0.1472), up from 0.0495. The matched comparison was understating the penalty by hobbling Pre-LN at a conservative rate. This is the cleanest demonstration of regime dependence in the study: same architectures, same data, same seeds, a 2.9-fold change in the headline number from one hyperparameter. **Third, QK-Norm's Post-LN-side contribution amplifies to ≈ 0.22 nats** once each side is at its own selected rate, against 0.0577 for the same single-axis step on the Pre-LN side—a 3.8-fold asymmetry—because on the Post-LN side QK-Norm improves the loss *and* unlocks the higher learning rate. The Post-LN-penalty pairs survive Bonferroni at $\alpha = 0.01$; the QK-Norm pairs, with d of 8–10, clear only $\alpha = 0.05$.

Block 2 carries the legacy closure and QKNorm-PostLN's selected 10^{-3} is outside the regime Block 7 measured, so Block 9 re-trains that margin canonically: PreLN − QKNorm-PostLN is $+0.0763$ nats ($+0.0670$ to $+0.0857$) against $+0.0755$ in Block 2, a difference of $+0.0009$, with the closure moving QKNorm-PostLN by -0.0009 nats (largest shift 0.0016). Its PreLN cells, for which the closure is a no-op, came back bit-identical to Block 2's, so this comparison carries no block-to-block offset at all (Appendix B).

4.2 Attributing the Post-LN rescue: a single-architecture chain

QKNorm-PostLN differs from PostLN on two axes—normalisation function and QK-Norm—and Block 6 was added to separate them. Because Block 6 is canonical-closure and Block A legacy, the chain uses the Block 7 end-points, so all three rows share architecture, learning rate, horizon and seeds (Table 3).

Swapping LayerNorm for RMSNorm on the Post-LN side moves the loss by $+0.0001$ nats, paired range -0.0012 to $+0.0009$. That ± 0.001 spread is small in absolute terms, not merely relative to noise: it is the tightest paired comparison in the study, because these runs share seeds *and* architecture. Adding QK-Norm at fixed RMSNorm then moves the loss by -0.0504 nats (0.0452–0.0531, $d = 11.2$): essentially the entire 0.050-nats closure from PostLN to QKNorm-PostLN is the QK-Norm step. The cross-block versions of the same two steps give the same answers with $14\times$ and $3.5\times$ the paired spread (Appendix D). QK-Norm is thus the load-bearing axis on the Post-LN side

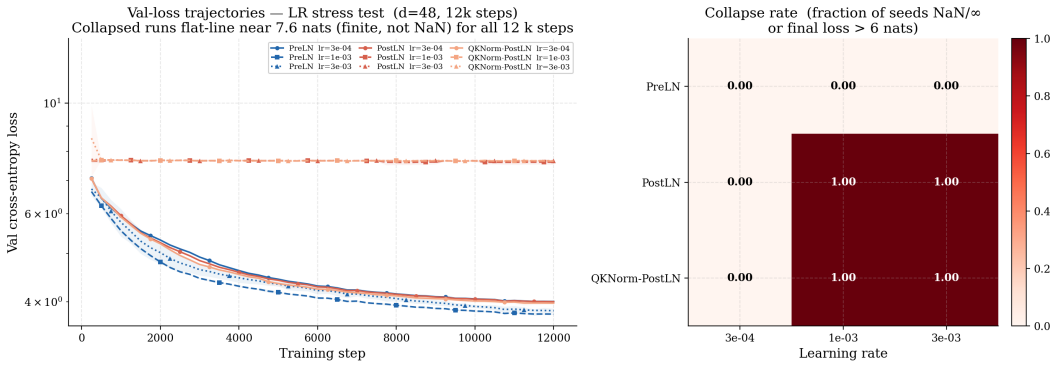


Figure 2: Block 3, depth 48, clip $= 100$, 12 k steps, three seeds. Left: validation-loss trajectories per (variant, LR), on a logarithmic loss axis; the Post-LN family at LR $\geq 10^{-3}$ is at ≈ 7.6 nats by the first evaluation—one run overshoots to 10.1 at step 250 before settling there—and stays for all 12 k steps, while Pre-LN keeps descending. Right: collapse rate per cell.

Table 4: Gradient clip against learning rate at depth 48, for the two Post-LN variants. Reading down a column isolates the learning rate; reading across a row isolates the clip. Cells give the number of runs (two variants $\times$ their seeds) that survive or collapse, the range of final losses, and the step budget of the block they come from (Block 4 at 20 k, Block 3 at 12 k, Block 8 at 1 k). "Collapse" is a final loss above 6 nats.

Peak LR	clip $= 1.0$	clip $= 100$
3×10^{-4}	survives 4/4, 3.728–3.767 (20k)	survives 6/6, 3.955–4.002 (12k)
10^{-3}	***collapse* 6/6, 7.66–7.68 (1k)**	*collapse* 6/6, 7.52–7.67 (12k)

both at fixed learning rate and for learning-rate tolerance; RMSNorm contributes nothing detectable to either.

4.3 Depth 48: parity below the ceiling, collapse above it

At 20 k steps, LR $= 3 \times 10^{-4}$ and clip $= 1.0$, the five variants run at depth 48 land within 0.021 nats of one another (3.7388–3.7593)—about two standard errors of a variant mean at $n = 2$, so we read it as "not statistically distinguishable", not as equality. The classical Post-LN penalty is not visible in this converged, conservatively-optimised cell (Appendix E). Block 3 shows what it takes to bring it back: with the clip relaxed to 100, QK-Norm does *not* rescue Post-LN at depth 48—QKNorm-PostLN and PostLN collapse together, $3/3$ seeds each, at every learning rate $\geq 10^{-3}$, to the same plateau, while Pre-LN survives every setting (Figure 2).

Which variable is responsible. Block 3 moves the clip and the learning rate together, so on its own it cannot say which kills these runs. Block 8 supplies the missing cell: the same two variants at depth 48 and LR $= 10^{-3}$, with the clip returned to its default of 1.0 and every other setting—closure, warm-up, seeds—held to Block 3's. All 6 runs collapse to the same plateau as their clip-100 counterparts, and the clip was engaged in every one (interval-mean pre-clip norm peaking at 1.01–3.11): it fired, and the runs died anyway. Meanwhile at 3×10^{-4} nothing collapses at either clip setting (Table 4). **Of the two variables crossed, the learning rate determines the depth-48 outcome and the clip does not**, as Block 1 also showed at depth 12. The factorial licenses the attribution between these two variables, not a mechanism: optimiser state, warm-up schedule and Post-LN layout all mediate how the failure unfolds.

This refines what QK-Norm buys. The threefold ceiling lift it wins at depth 12 is gone at depth 48: QKNorm-PostLN collapses at 10^{-3} exactly as vanilla Post-LN does, clipped or not. **QK-Norm's learning-rate headroom does not survive to depth 48**, bounding the combined variant's selected-LR advantage to the depth measured. Our grid steps by half-decades, so what is established is that QK-Norm's depth-48 ceiling has fallen below 10^{-3}—into the same grid interval as vanilla Post-LN's—not that the two ceilings coincide. Blocks 3 and 8 both carry the legacy closure; Block 10 re-

runs Block 8 with the closure flipped and every other setting held, and all 6 of the 6 runs again finish above the threshold: the 5 that reached the plateau sit within 0.0508 nats of their legacy counterparts, and the sixth is the one seed still descending at the end of the 1 k screen (Appendix B). The depth-48 result is a property of the canonical architecture.

What the collapse is. It is tempting to call these runs diverging, but the artefacts say something more specific. The validation loss is already ≈ 7.69 nats at the first evaluation—step 250, inside the 500-step warm-up—and never moves; the interval-mean pre-clip gradient norm is 1.07–1.58 over the first interval against 1.11–1.12 for Pre-LN, then falls to 0.17–0.38 and stays. Uniform prediction over $50{,}257$ tokens would give 10.82 nats, so the plateau is not uniform but almost exactly the unigram solution. The maximum-likelihood token entropy of the validation construction is 7.641 nats, and of the 29 runs finishing above the collapse threshold across Blocks 1, 3, 6, 8 and 10, the 28 that have settled average 7.661, i.e. $+0.020$ nats above it. Wang et al. (2026) derive the same endpoint independently; we add the measured entropy to place it against, and that neither QK-Norm nor the clip prevents it. The model has collapsed to the marginal token distribution and has too little gradient to leave it: an early warm-up trap with vanishing gradients, not a late blow-up.

4.4 Mechanism, DeepNet and throughput

Figure 9 (Appendix F) shows per-block pre-clip gradient norms at depth 24 over the 5 k-step window all variants share. The Pre-LN family shows the expected gradient-highway picture: smooth across depth, roughly flat over training, mean $\log_{10} \|g\|$ of -1.15. PostLN is higher everywhere (-1.05): each norm on the residual stream contributes a Jacobian factor in the backward pass. QK-Norm attenuates the Post-LN side toward the Pre-LN range (-1.07) without changing the residual layout, and its last/first ratio (0.264) is essentially PostLN's (0.267); the layout, not the attention logits, governs how gradient magnitude is distributed across depth. DeepNet's α/β scaling damps magnitude more than any variant (-1.22) and places its ratio (0.211) between the two families (Appendix F, Table 17). The clip's action is located precisely: from depth 12 upward PostLN is the only variant whose interval-mean pre-clip norm exceeds 1.0 after warm-up, and the frequency falls as the model deepens—40% of intervals at depth 12, 20% at 24, 12% at 48, all at the 5 k horizon—while every other cell sits at the floor that corresponds to the first interval of each run and nothing else (Table 18). So at the depth where relaxing the clip collapses Post-LN, the clip is engaged almost only during warm-up—the gradient-level counterpart of the warm-up trap above.

DeepNet and throughput. DeepNet did not help in this regime: it sits 0.07 nats above vanilla PostLN at depth 12 and 0.11–0.13 above the converged band at depth 48, though it *does* converge at every depth without QK-Norm and without any learning-rate tuning, which is its core promise. We report this as a regime-specific observation and not a result about DeepNet: its learning rate was never swept, its initialisation differs from PostLN's on every layer, and a clip of 1.0 at $d \leq 48$ already absorbs what α/β would (Appendix F, Table 19). On throughput (Appendix F, Figure 11), LayerNorm $\rightarrow$ RMSNorm is $+3.2\%$ and Pre-LN $\rightarrow$ Post-LN is neutral (-0.1%), as it should be at equal operation count; adding QK-Norm costs 8.2% in latency in an unfused implementation, so RMSNorm + QK-Norm runs 4.6% slower than the LayerNorm Pre-LN baseline—a small price against the 0.0602–0.0755 nats the QK-Norm variants gain at selected learning rates, though the figure is implementation-specific and we do not generalise it.

5 Discussion and conclusion

Ask which regime, not which architecture. The most defensible result here is not any single ordering but the demonstration that the ordering moves. Blocks A and 2 compare the same architectures on the same data with the same seeds and differ by $2.9\times$ in the headline Post-LN penalty, moving only the learning rate; Blocks 4 and 3 differ between parity and total collapse at depth 48. Neither pair changes the model. Any normalisation comparison that fixes a single learning rate, horizon and clip is measuring the interaction of the architecture with those choices as much as the architecture itself.

What QK-Norm buys, and what RMSNorm does not. QK-Norm does two separable things: at a fixed learning rate it lowers the loss by ≈ 0.03 nats on the Pre-LN side and ≈ 0.05 on the Post-

LN side, and it raises Post-LN's learning-rate ceiling by $3\times$, which is where most of the combined variant's selected-LR advantage comes from. Both are consistent with the attention-logit-growth account of Dehghani et al. (2023) and Wortsman et al. (2024). What it does *not* do is substitute for the residual layout: its gradient distribution across depth is Post-LN's, and at depth 48 it collapses exactly as vanilla Post-LN does—it fixes the attention pathway, not the residual pathway. RMSNorm, meanwhile, matches LayerNorm within the paired noise on both sides ($+0.0008$ sign-changing across seeds; $+0.0001$, ± 0.001) and leaves Post-LN's low ceiling exactly where it was: the "drop-in" folk theorem holds for quality and speed, not stability.

Post-LN's instability is a warm-up trap. The collapses in Blocks 1, 3, 6, 8 and 10 are of one kind: the loss reaches the corpus unigram entropy within 250 steps—in every case but the one seed of Appendix B—and the gradient then vanishes. That is the early-gradient mechanism of Xiong et al. (2020), but its phenomenology—finite, flat, quiet—is different from the loss spikes usually pictured, and it explains why neither clip setting matters at depth 48. The collapse is complete inside 250 steps and the clip is engaged there and almost nowhere else; clipping bounds the *size* of those first updates but not their direction. A rail that only acts during the window in which the damage is done, and acts by rescaling rather than redirecting, is not a rail.

For practitioners, in this regime. At ≈ 21 M non-embedding parameters, depth ≤ 48, AdamW, clip $= 1.0$, one learning rate per run and no μP: RMSNorm + QK-Norm + Pre-LN is the safe default; RMSNorm + QK-Norm + Post-LN at LR $\approx 10^{-3}$ reaches the lowest loss at depth 12, but do not carry that learning rate to depth 48, where it collapses at either clip setting we tried; vanilla Post-LN buys a learning-rate ceiling too low to be worth it at any depth; RMSNorm is quality-neutral here and 3.2% faster in our implementation. These are candidates to test in one's own setting.

In sum. Tested together, the three folk theorems each survive in a qualified form. Pre-LN is more stable than Post-LN—once the learning rate is raised; at a conservative rate the two reach parity at depth 48. RMSNorm is quality-neutral on both sides of the layout axis and faster in our implementation, but stabilises nothing. QK-Norm stabilises attention enough to raise Post-LN's ceiling threefold and make the combined variant the best depth-12 model at selected rates, but that headroom is depth-limited. The main export we would like is the design: single-axis chains, exact seed pairing, learning rate and clip treated as variables, and every artefact released.

6 Limitations

Appendix G gives the full list and the follow-ups we would run first.

- **Three seeds, and what that buys.** Effect sizes and paired ranges are the evidence; p-values are not. Every directional claim holds in all three seeds, but only 1 matched-block pair and the four Post-LN-penalty pairs of the selected-LR block survive Bonferroni at $\alpha = 0.01$, and the QK-Norm claims only at $\alpha = 0.05$ (Section 3.3).
- **Learning rates were chosen at one seed and one horizon, then applied at another.** Block 1 selects at 5 k steps; Block 2 trains for 20 k. Optimal learning rate generally falls as the horizon grows (Hägele et al., 2024), so the Pre-LN picks at the top of the grid are the likeliest to be too hot at 20 k, which would inflate PreLN's measured loss and therefore *understate* the selected-LR Post-LN penalty. There is also no held-out test split.
- **The Post-LN closure changed mid-study, and cross-block comparisons carry an offset.** The change is -0.0001 nats (Table 1 gives the closure per block), the headline attribution does not depend on it, and Blocks 9 and 10 re-measured the two claims outside that regime without either moving; depth-48 parity is the one claim still covered by extrapolation alone. Separately, the same configuration and seed run in Blocks A and 2 land up to 0.018 nats apart—an offset shared by every run in a block, not per-run noise—which affects the RMSNorm-PostLN rows and the DeepNet comparisons.
- **Single width, one corpus, depth ≤ 48, no μP, two seeds at depth 48.** Learning-rate optima do not transfer across width without μP (Yang et al., 2021), so orderings taken at selected rates may shift at other widths (Kim et al., 2025; Li et al., 2025). DeepNet is separately confounded.

REPRODUCIBILITY STATEMENT

The repository accompanying this paper contains the training code (PyTorch (Paszke et al., 2019), no trainer framework), the run specifications for all thirteen blocks, the per-run artefacts (`config.json`, `metrics.json`, `summary.json`) for all 175 runs, and an analysis pipeline that regenerates every table and figure from those artefacts with one command; no GPU is required to reproduce any number in this paper. The tables are themselves generated from the pipeline's outputs by a script in the paper directory, and every headline number quoted in the prose is a macro emitted by that script rather than a typed literal. Each run's configuration records which Post-LN closure it used, and a script re-derives that flag from the recorded parameter counts as a check. Package versions are pinned to those recorded for the committed runs (PyTorch 2.11, CUDA 13.0). Training itself is bit-reproducible on this hardware: Block 11 re-runs two configurations in two separately launched studies and all 8 comparisons agree to the last representable digit at every evaluation. Two caveats remain: comparisons that cross a block are not exactly reproducible (Blocks A and 2 differ by up to 0.018 nats at a fixed seed, an offset between blocks rather than per-run noise), and the 19.7 MB validation file is not committed, though its construction is deterministic and its SHA-256 is recorded for runs from the current code onward. An anonymised copy of the repository, including all per-run artefacts and the analysis pipeline, is available to reviewers at `https://anonymous.4open.science/r/gpt-from-scratch-ablations-5946`.

AI USE STATEMENT

AI assistants (Anthropic's Claude, via Claude Code) were used throughout this work. They drafted and revised prose in every section of this paper under the author's direction; wrote and reviewed the analysis and table-generation code; specified the run configurations of the later study blocks from the author's design decisions; and performed repeated audits of the manuscript against the committed run artefacts, which found and corrected numerical and consistency errors. They also searched the literature for related and concurrent work and checked every bibliography entry against a publisher, arXiv, proceedings or Crossref record; the author confirmed the relevance of each citation and the claim attributed to it. All experiments were launched by the author; every number in the paper is generated from the run artefacts by the released pipeline; the author reviewed every claim and takes full responsibility for the content.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Chen Chen and Lai Wei. Post-LayerNorm is back: Stable, ExpressivE, and deep. *arXiv preprint arXiv:2601.19895*, 2026.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 2nd edition, 1988.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7480–7512, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, 2010.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 76232–76264, 2024. doi: 10.52202/079017-2427.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. In *International Conference on Learning Representations*, 2017.

Jeonghoon Kim, Byeongchan Lee, Cheonbok Park, Yeontaek Oh, Beomjun Kim, Taehwan Yoo, Seongjin Shin, Dongyoon Han, Jinwoo Shin, and Kang Min Yoo. Peri-LN: Revisiting normalization layer in the transformer architecture. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30400–30436, 2025.

Pengxiang Li, Lu Yin, and Shiwei Liu. Mix-LN: Unleashing the power of deeper layers by combining pre-LN and post-LN. In *International Conference on Learning Representations*, 2025.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5747–5763, 2020.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Toan Q. Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*, 2019.

OLMo Team. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2025.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1310–1318, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8024–8035, 2019.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, pp. 30811–30849, 2024.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, 2017.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Oleg Rybakov, Mike Chrzanowski, Peter Dykas, Jinze Xue, and Ben Lanir. Methods of improving LLM training stability. *arXiv preprint arXiv:2410.16682*, 2024.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725, 2016.

Sam Shleifer, Jason Weston, and Myle Ott. NormFormer: Improved transformer pretraining with extra normalization. *arXiv preprint arXiv:2110.09456*, 2021.

Wenfang Sun, Xinyuan Song, Pengxiang Li, Lu Yin, Yefeng Zheng, and Shiwei Liu. The curse of depth in large language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 163104–163136, 2025. doi: 10.52202/085713-5446.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. Spike no more: Stabilizing the pre-training of large language models. *arXiv preprint arXiv:2312.16903*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6761–6774, 2024. doi: 10.1109/TPAMI.2024.3386927.

Xingjian Wang, Qingyu Han, Xiaodong Luo, and Yin Zhang. Why post-norm transformers collapse: Attention amplification and gradient repair failure. *arXiv preprint arXiv:2608.09417*, 2026.

Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D. Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-Dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities. In *International Conference on Learning Representations*, 2024.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10524–10533, 2020.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17084–17097, 2021.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M. Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40770–40803, 2023.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12360–12371, 2019.

Zhijian Zhuo, Yutao Zeng, Ya Wang, Sijun Zhang, Jian Yang, Xiaoqing Li, Xun Zhou, and Jinwen Ma. HybridNorm: Towards stable and efficient transformer training via hybrid normalization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 25106–25147, 2025. doi: 10.52202/085713-0844.

A EXPERIMENTAL DETAILS

Table 5: The six single-axis variants and the DeepNet baseline. The Pre-LN and Post-LN chains each add one axis per step; every headline pairwise comparison is between rows that differ on exactly one column.

Variant	Norm	Layout	QK-Norm	Init	Isolates (vs. previous row in chain)
PreLN	LayerNorm	Pre-LN	–	GPT-2	GPT-2 baseline
RMSNorm-PreLN	RMSNorm	Pre-LN	–	GPT-2	normalisation function
QKNorm-PreLN	RMSNorm	Pre-LN	✓	GPT-2	attention-logit normalisation
PostLN	LayerNorm	Post-LN	–	GPT-2	layout (vs. PreLN)
RMSNorm-PostLN	RMSNorm	Post-LN	–	GPT-2	normalisation function
QKNorm-PostLN	RMSNorm	Post-LN	✓	GPT-2	attention-logit normalisation
DeepNet-PostLN	LayerNorm	Post-LN	–	DeepNet α/β	initialisation-based rescue

Parameter counts. RMSNorm drops the per-channel shift and QK-Norm adds two 64-dimensional scale vectors per block, which is why the variants sit within 0.05% of the $21{,}280{,}512$ non-embedding parameters of the LayerNorm variants at $N = 12$. Post-LN counts quoted in the main text are for the legacy closure of Section 3.3; the canonical closure removes one further norm ($21{,}279{,}744$ at $N = 12$), which is how each run's closure is re-derivable from its recorded parameter count. Token and position embeddings add a further 19.5 M. Non-embedding count scales linearly with depth: 10.6 M, 42.6 M and 85.1 M at $N = 6, 24, 48$.

Evaluation. The number we report is the value at the *last* evaluation, at exactly `max_steps`—not the minimum over training and not an average of late evaluations. Section 3.3 quantifies the sampling noise this introduces and why it cancels in paired comparisons.

Optimisation. Warm-up is 200 steps for 5 k-step runs and 500 for longer runs and for the 1 k-step Block 8, which keeps Block 3's warm-up by design. Dropout is zero everywhere. A 5 k-step run sees 164 M tokens. The gradient norm is recorded before clipping, both globally at every step and per block every 25 steps where per-layer logging is enabled.

B FINAL-NORM CLOSURE CALIBRATION

Table 6 gives the per-seed and aggregate result of Block 7: the two Post-LN cells of Block A retrained under the canonical closure, everything else identical, paired seed for seed against the originals.

Table 6: Legacy (`ln_f`) vs. canonical (`Identity`) Post-LN closure, depth 12, LR $= 3 \times 10^{-4}$, 20 k steps.

Variant	Seed	Legacy (`ln_f`)	Canonical (`Identity`)	Δ = canonical − legacy
PostLN	0	3.9963	3.9783	-0.0179
PostLN	1	3.9999	4.0129	+0.0129
PostLN	2	3.9801	3.9857	+0.0057
QKNorm-PostLN	0	3.9426	3.9338	-0.0088
QKNorm-PostLN	1	3.9581	3.9586	+0.0004
QKNorm-PostLN	2	3.9230	3.9336	+0.0106
PostLN (mean of 3)				**+0.0002** (sd 0.0161, 95% CI $\pm$0.0401, $p = 0.98$)
QKNorm-PostLN (mean of 3)				**+0.0007** (sd 0.0097, 95% CI $\pm$0.0242, $p = 0.91$)

The mean shifts are $+0.0002$ and $+0.0007$ nats and the per-seed shifts straddle zero, but their spread (up to 0.018 nats) is not the closure's doing. Block A and Block 7 were launched separately, and a separate launch is itself worth 0.018 nats at a fixed seed. Table 7 takes the two apart using the one cell that exists three times: PostLN at depth 12 and LR $= 3 \times 10^{-4}$, trained in Block A and again in Block 2 under the legacy closure, and in Block 7 under the canonical one, with Blocks 2 and 7 pairing at $r = 0.99995$ despite being launched three days apart.

The closure itself moves the loss by -0.0001 nats, with no seed shifted by more than 0.0002 and a 95% interval of ± 0.0003—two orders of magnitude below the effects it is being asked to bound,

Table 7: The closure change separated from the launch effect, using the PostLN cell at depth 12, LR $= 3 \times 10^{-4}$, 20 k steps, three paired seeds. Blocks 2 and 7 differ only in the closure and pair at $r = 0.99995$; Blocks A and 2 share the closure and differ only in the block, and do not pair ($r \approx 0$).

Comparison	Isolates	$\bar{\Delta}$	sd	max $\lvert\Delta\rvert$	p
Block 2 → Block 7	final-norm closure	-0.0001	0.0001	0.0002	0.40
Block A → Block 2	launch, config fixed	+0.0003	0.0162	0.0179	0.98
Block A → Block 7	closure + launch	+0.0002	0.0161	0.0179	0.98

and consistent with $\mathrm{LN}(\mathrm{LN}(x))$ differing from $\mathrm{LN}(x)$ only through the channel statistics of the inner norm's learned scale. Blocks A and 2, by contrast, differ by 0.018 nats at a fixed seed (s.d. 0.016) despite sharing a configuration and a closure—an offset between the blocks, not launch nondeterminism (Appendix C). The naive Block A-versus-Block 7 comparison is the sum of the two and is dominated by the second. Table 8 sets out which downstream claims this calibration bounds directly and which it does not.

Table 8: What the calibration does and does not bound.

Claim	Covered?
Matched-LR Post-LN penalty, $+0.0495$ (§4.1)	Directly: same cell, same LR; a -0.0001 shift on a 0.05 effect.
Single-axis chain (§4.2)	Not needed: entirely within one architecture and one launch.
Selected-LR margin, 0.0755 (§4.1)	Directly, by Block 9: the same cell re-trained canonically at 10^{-3}, margin $+0.0763$ vs $+0.0755$.
Depth-48 parity within 0.021 (§4.3)	By extrapolation only: measured at depth 12.
Depth-48 collapse (§4.3)	Directly, by Block 10: Block 8 re-run canonically, $6/6$ still collapse.
DeepNet (§4.4)	Not covered: different initialisation scheme.

Given a measured closure effect of 10^{-4} nats against effects of 0.05–0.14, re-running the 73 legacy Post-LN cells is not warranted. A 12-run extension was the proportionate check instead, and Blocks 9 and 10 are it: they re-trained the two claims that lay outside the regime this calibration was measured in. That distinction is worth stating, because it is why they were needed. The calibration was measured on a *converged loss* at a conservative learning rate, whereas the depth-48 rows concern a collapse that completes inside the first 250 steps, and the redundant norm changes the Jacobian into the last block as well as the forward pass. Carrying a 10^{-4} steady-state effect into a question about early-training dynamics would have been an assumption rather than a measurement. Both re-runs agreed with the extrapolation—the selected-LR margin moved by $+0.0009$ nats and the depth-48 collapse not at all—so the remaining row reads "by extrapolation only" because depth-48 parity was not re-run, not because it is doubted.

The one unsettled run. Under the canonical closure, PostLN seed 2 of Block 10 had not reached the plateau by the end of the screen: it ends at 6.29 nats and is still descending (-0.57 nats over the last two evaluations, against -0.04 to $+0.06$ for the 5 that are flat). Its legacy counterpart took a gradient spike at step 500 and settled on the plateau; the canonical run did not spike. The two agree to 0.013 nats at the first evaluation, inside the block-to-block variation of Section 3.3, so we read this as a marginal seed near the boundary of the collapse basin rather than as an effect of the closure—but at a 1 k horizon we cannot say which side it would have ended on, and we have not run it further. It is the one place in this study where the claim "every collapse is complete by the first evaluation" does not hold, and it is the run excluded from the plateau statistic of Section 4.3. The clip relaxation is also not quite a no-op for Pre-LN: its interval-mean pre-clip norm averages 0.36–0.71 but reaches 1.46 in its worst window at 3×10^{-4}, above the default clip, so those runs are genuinely unclipped where a default run would occasionally not have been.

C STATISTICAL DETAILS

Multiple comparisons. Table 9 counts, per block, how many of the pairwise comparisons pass at each threshold, uncorrected and Bonferroni-corrected over the pairs in that block.

Table 9: Pairwise paired t-tests per block, uncorrected and Bonferroni-corrected.

Block	Pairs	Uncorrected $p < 0.01$	Bonferroni $p < 0.01$	Bonferroni $p < 0.05$
A — matched LR	10	6	**1**	4
2 — selected LR	10	8	**4**	8

The four survivors in the selected-LR block are the comparisons of PostLN against each of the other four variants. In the matched block 1 of the 10 pairs survives at $\alpha = 0.01$—QKNorm-PreLN vs. PostLN—and 4 do at 0.05: PostLN against each of PreLN, RMSNorm-PreLN and QKNorm-PreLN, and RMSNorm-PreLN vs. QKNorm-PreLN.

The family is the 10 comparisons among the five variants Block A actually trained together. The five pairs involving RMSNorm-PostLN, which comes from Block 6, are excluded: those runs do not share validation batches with Block A's ($r \approx 0$ against $r = 0.995$ within a block), so a paired t-test on them assumes a pairing that is not there. We report their differences and ranges and withhold d and p, rather than correcting them as though they were tests. Including them would have divided by 15 instead of 10, which is stricter and so cannot inflate a claim—but a paired statistic on unpaired runs is the wrong quantity whichever way the correction cuts. The single-architecture chain of Table 3 is what carries the RMSNorm-PostLN attribution formally.

Validation-sampling noise. For each of the 33 depth-12, 20 k-step runs of Blocks A, 2 and 6 we fitted a line to the last eight validation evaluations and took the residual standard deviation as an estimate of per-evaluation noise: median 0.009 nats.

Pairing verification. Constructing each of the seven variant models at a fixed seed and hashing the global `torch` RNG state afterwards gives the same hash for all 14 variant-and-closure combinations; the data shuffle is seeded from the run seed and the `DataLoader` draws its base seed from the same global state at every evaluation. Within a block this makes the pairing exact, and it is visible in the residuals: the variants of a block at a shared seed correlate at $r = 0.995$ pairwise, confirming that they sample the same validation batches at the same steps.

Across blocks they do not. The same configuration and seed, run in Blocks A and 2, gives residuals that are uncorrelated ($r \approx 0$) and final losses up to 0.018 nats apart (s.d. 0.016 over three seeds; Table 7), so a paired difference taken across those blocks carries the full per-run variation instead of cancelling it.

It is worth being precise about the cause, because the obvious explanation is the wrong one. Training is bit-reproducible on this hardware. Block 11 runs two configurations twice under different run ids in two separately launched studies: all 8 comparisons—4 across containers inside one launch, 4 across the launch boundary—agree at every evaluation to the last representable digit, with a largest absolute difference in final loss of exactly zero. Block 9 corroborates this at the full 20 k horizon: its PreLN cells, for which the closure is a no-op, reproduce Block 2's PreLN cells exactly, months apart. The scope of that evidence is depth 12 on H100 within one container image, for runs that complete without a restart; we have not replicated a depth-48 run on H200, and long runs that resume from a checkpoint restore RNG state through a separate path we have not tested this way.

Two things follow, and it is worth separating them. The first is secure: the Block A/Block 2 spread is not floating-point non-determinism, and no determinism flag would remove it. Because two runs of one configuration are bit-identical, whatever separates the two blocks is an *input* to the block rather than per-run randomness—and every run in a block shares its inputs, so the difference is a common-mode offset rather than independent noise. Independent noise shrinks when averaged over seeds; an offset does not. Treating 0.018 nats as a bound on cross-block comparisons, rather than as a standard error to be divided by $\sqrt{n}$, is the conservative reading and the one this paper takes. That conclusion holds whichever input turns out to be responsible.

The second is not secure, and we state it as a candidate rather than a finding. The two blocks record identical package versions, Python version and GPU model; their data-loading code is unchanged between them; and their trajectories disagree at the *first* evaluation, before numerical error could accumulate. The unpinned FineWeb-Edu stream is the natural explanation, but the runs of that era recorded no dataset revision and no digest of the validation file, so we cannot demonstrate it after the fact, nor exclude a rebuild of the validation set in the same window. This is why the two cross-block rows of Table 10 show $14\times$ and $3.5\times$ the paired spread of their within-block counterparts in Table 3, and why the attribution in Section 4.2 is made inside Blocks 6–7, which were launched within hours of each other and whose residuals correlate at $r = 0.998$ (minimum 0.985).

Bootstrap. The released significance tables also contain a percentile bootstrap interval for each mean paired difference (10,000 resamples). With $n = 3$ its support is degenerate (≈ 10 distinct resample sums) and it reduces to the min–max range, which is why we report the range directly.

D DEPTH-12 RESULTS IN FULL

Tables 10 and 11 give the paired statistics behind Section 4.1, and Table 12 the learning-rate grid behind Figure 1.

Table 10: Paired comparisons in the matched block. $\bar{\Delta} = \mathrm{loss}(a) - \mathrm{loss}(b)$; range is the exact min–max over three seeds; p is the paired t-test and p_{Bonf} its correction over the 10 genuinely paired comparisons in the block, of which the seven that isolate a single axis or a headline contrast are shown. ‡Cross-block, cross-closure comparison (Block 6 vs. Block A): these two rows do not share validation batches and carry the offset quantified in Section 3.3, so their difference and range stand but d and p are withheld and they are not members of the Bonferroni family; the single-architecture, single-launch version is in Table 3.

Pair ($a - b$)	Axis isolated	$\bar{\Delta}$	Δ range	Cohen's d	p	p_{Bonf}
QKNorm-PreLN − PreLN	QK-Norm, Pre-LN side	-0.0284	[-0.0320, -0.0237]	-6.7	0.007	0.07
RMSNorm-PreLN − PreLN	LN → RMSNorm, Pre-LN side	+0.0008	[-0.0023, +0.0035]	+0.3	0.67	1.00
PostLN − PreLN	Layout	+0.0495	[+0.0447, +0.0535]	+11.1	0.003	0.03
QKNorm-PostLN − PreLN	Combined vs. baseline	-0.0013	[-0.0124, +0.0087]	-0.1	0.85	1.00
QKNorm-PostLN − QKNorm-PreLN	Layout, given QK-Norm	+0.0271	[+0.0172, +0.0407]	+2.2	0.06	0.61
RMSNorm-PostLN − PostLN	LN → RMSNorm, Post-LN side‡	+0.0003	[-0.0173, +0.0117]	–	–	–
QKNorm-PostLN − RMSNorm-PostLN	QK-Norm, Post-LN side‡	-0.0512	[-0.0636, -0.0364]	–	–	–

Table 11: Paired comparisons in the selected-LR block (10 pairs; p_{Bonf} corrected over all ten).

Pair ($a - b$)	Axis isolated	$\bar{\Delta}$	Δ range	Cohen's d	p	p_{Bonf}
QKNorm-PostLN − PreLN	Combined vs. baseline	-0.0755	[-0.0848, -0.0670]	-8.5	0.005	0.0462
QKNorm-PreLN − PreLN	QK-Norm, Pre-LN side	-0.0602	[-0.0671, -0.0563]	-10.1	0.003	0.03
RMSNorm-PreLN − PreLN	LN → RMSNorm, Pre-LN side	-0.0025	[-0.0036, -0.0008]	-1.7	0.10	0.99
PostLN − PreLN	Layout	+0.1442	[+0.1413, +0.1472]	+48.8	<0.001	0.001
QKNorm-PostLN − QKNorm-PreLN	Layout, given QK-Norm	-0.0153	[-0.0177, -0.0107]	-3.9	0.02	0.22
QKNorm-PostLN − PostLN	Full Post-LN rescue	-0.2197	[-0.2260, -0.2141]	-36.7	<0.001	0.002

Table 12: Block 1 (+ Block 6 row): final validation loss vs. peak learning rate, depth 12, 5 k steps, one seed per cell. Bold marks each variant's best value in the sweep, which sets its learning rate in Block 2. "Collapse" = final loss above 6 nats.

Variant	1×10^{-4}	3×10^{-4}	1×10^{-3}	3×10^{-3}	Selected LR
PreLN	5.388	4.696	4.246	**4.240**	3×10^{-3}
RMSNorm-PreLN	5.395	4.696	4.249	**4.218**	3×10^{-3}
QKNorm-PreLN	5.352	4.618	**4.266**	4.277	1×10^{-3}
PostLN	5.522	**4.677**	*collapse*	*collapse*	3×10^{-4}
RMSNorm-PostLN	5.526	**4.676**	*collapse*	*collapse*	3×10^{-4}
QKNorm-PostLN	5.542	4.599	**4.120**	*collapse*	1×10^{-3}

The full per-variant tables are Tables 13 and 14; Table 2 in the main text is the two of them side by side, without the perplexity and throughput columns.

Table 13: Block A (+ Block 6†): depth 12, LR $= 3 \times 10^{-4}$, 20 k steps, three seeds. Val PPL is exp of the mean loss; throughput is tokens/wall-clock over the full run, including evaluation, on H100.

Variant	Final val loss	Val PPL	Δ vs PreLN	Throughput (tok/s)
PreLN	3.9426 ± 0.0070	51.55	–	92,371
RMSNorm-PreLN	3.9434 ± 0.0057	51.59	+0.0008	95,338
QKNorm-PreLN	$\mathbf{3.9141 \pm 0.0073}$	50.10	-0.0284	88,121
PostLN	3.9921 ± 0.0106	54.17	+0.0495	92,256
RMSNorm-PostLN†	3.9924 ± 0.0171	54.19	+0.0499	94,427
QKNorm-PostLN	3.9413 ± 0.0176	51.48	-0.0013	89,336

Table 14: Block 2: depth 12, 20 k steps, three seeds, each variant at its sweep-selected learning rate.

Variant	Selected LR	Final val loss	Val PPL	Δ vs PreLN
PreLN	3×10^{-3}	3.8482 ± 0.0211	46.91	–
RMSNorm-PreLN	3×10^{-3}	3.8458 ± 0.0204	46.79	-0.0025
QKNorm-PreLN	1×10^{-3}	3.7880 ± 0.0153	44.17	-0.0602
PostLN	3×10^{-4}	3.9924 ± 0.0183	54.18	+0.1442
QKNorm-PostLN	1×10^{-3}	$\mathbf{3.7728 \pm 0.0125}$	43.50	-0.0755

Three findings from the sweep, in full. First, the Pre-LN family is robust across the whole grid, with its best values at the top end; the difference between 10^{-3} and 3×10^{-3} is 0.006 nats for PreLN and 0.031 for RMSNorm-PreLN, inside what a single seed can resolve, so the Pre-LN-family picks could shift to the adjacent value with more seeds. Second, vanilla PostLN cannot tolerate LR $\geq 10^{-3}$ at depth 12: its best value is 3×10^{-4}, the very rate used in Block A, and above it the model collapses. Third, RMSNorm does not move that ceiling—RMSNorm-PostLN and PostLN are indistinguishable at single-seed resolution at every grid point and collapse at the same two—whereas QK-Norm moves it by $3\times$: QKNorm-PostLN survives 10^{-3} and, at that rate, records the lowest loss of any variant at any learning rate in the sweep (4.120 nats, 0.48 nats better than its next-best setting, so this pick is unambiguous).

Figure 3 and Figure 4 show the trajectories at matched and selected rates. Comparing the two shows the regime dependence as trajectories rather than endpoints: at the matched rate the six variants stay bunched for the whole run, while at selected rates PostLN separates from the rest within the first few thousand steps and never rejoins. Figure 5 gives the seed-level spread at both rates.

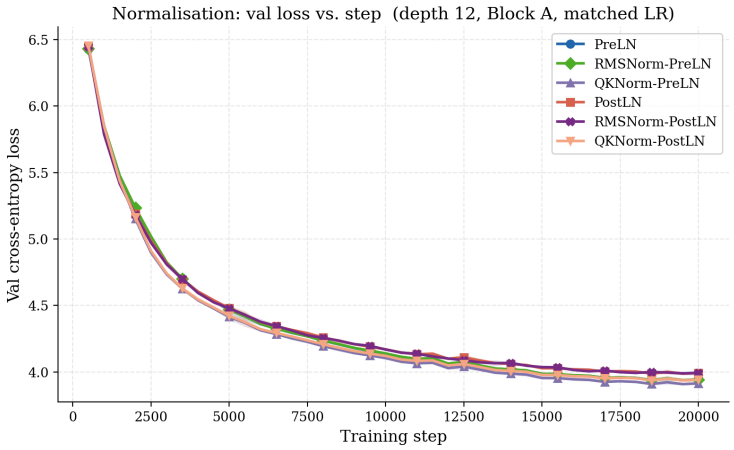


Figure 3: Validation loss over training at depth 12, matched LR (Block A + Block 6). Mean over three seeds with ± 1 s.d. band. The Pre-LN family bunches; the two Post-LN variants without QK-Norm trail together; QKNorm-PreLN leads.

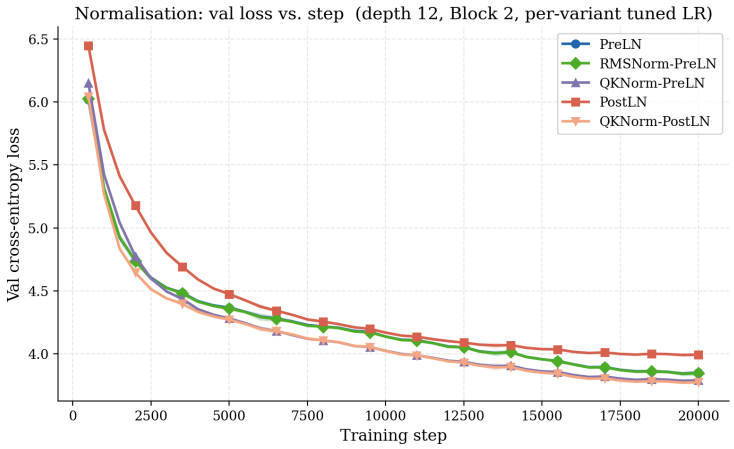


Figure 4: Validation loss over training at each variant's selected learning rate (Block 2). Compare Figure 3: the same variants at a shared learning rate, with RMSNorm-PostLN additionally present there.

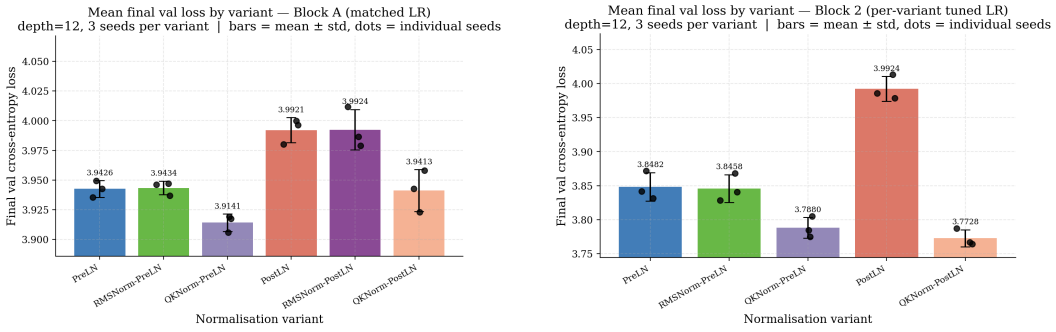


Figure 5: Final validation loss at depth 12, bars are mean ± s.d. with dots for individual seeds. **Left**: matched LR (Block A + Block 6); PostLN and RMSNorm-PostLN overlap to within seed spread, and QKNorm-PostLN is a clean step down. **Right**: per-variant selected learning rates (Block 2).

E DEPTH AND THE COLLAPSE

Table 15 and Figure 6 report Block B (depths 6, 12, 24 at 5 k steps, two seeds) alongside Block 4 (depth 48 at 20 k steps, two seeds). Block B also ran depth 48 at the short horizon; those cells are superseded by Block 4's converged ones and are not reported as losses, but they are the source of the depth-48 gradient diagnostics in Section 4.4 and Table 18. These are two different training horizons: the depth-24 and depth-48 columns are not comparable to each other, and this is not a scaling law. Read each column on its own.

Table 15: Final validation loss vs. depth at LR $= 3 \times 10^{-4}$. Depths 6–24 are 5 k-step runs (Block B, two seeds); depth 48 is 20 k steps (Block 4, two seeds). DeepNet is 20 k steps at every depth it was run at (Block 5, three seeds, depths 12–48); only its depth-48 cell shares a horizon with the column it sits in, which is why the other DeepNet cells are omitted here and given in Table 19. Columns are at different horizons and must not be read across.

Variant	$d = 6$ (5k)	$d = 12$ (5k)	$d = 24$ (5k)	$d = 48$ (**20k**)
PreLN	4.743 ± 0.027	4.672 ± 0.027	4.649 ± 0.008	3.758 ± 0.015
RMSNorm-PreLN	4.747 ± 0.028	4.674 ± 0.027	4.654 ± 0.004	3.759 ± 0.016
QKNorm-PreLN	4.652 ± 0.012	4.596 ± 0.006	4.581 ± 0.001	3.739 ± 0.014
PostLN	4.839 ± 0.004	4.667 ± 0.001	4.598 ± 0.008	3.759 ± 0.012
QKNorm-PostLN	4.754 ± 0.007	4.596 ± 0.021	4.493 ± 0.001	3.739 ± 0.015
DeepNet-PostLN (20k)	–	–	–	3.864 ± 0.013

At the early-training horizon, the QK-Norm variants lead at every depth, and the within-QK ordering shifts with depth: QKNorm-PreLN is ahead at $d = 6$, the two tie at $d = 12$, and QKNorm-PostLN pulls ahead at $d = 24$ by 0.09 nats—and by 0.16 nats over PreLN. We have no 20 k-step depth-24 run to say whether that margin is a final-loss effect or a training-speed effect.

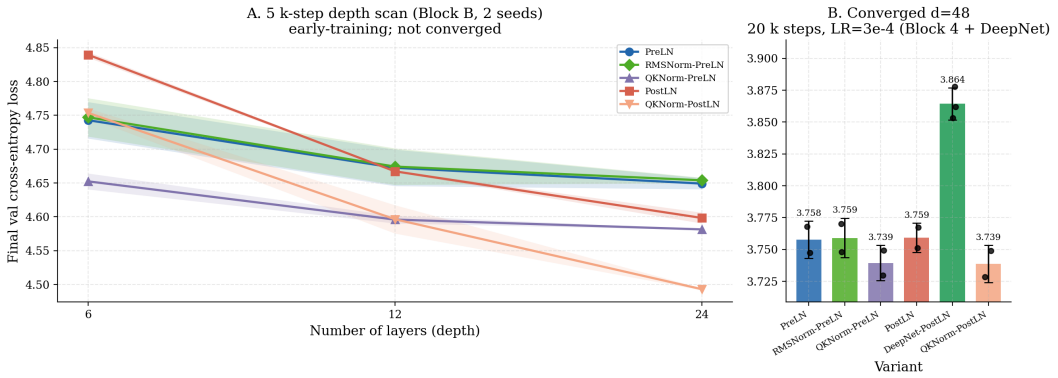


Figure 6: Two-horizon depth view, split on purpose. **A**: 5 k-step scan at $d \in \{6, 12, 24\}$ (Block B), well above the converged plateau. **B**: converged $d = 48$ at 20 k steps (Block 4 + DeepNet). Note the different y-ranges: at the converged depth-48 spot-check the inter-variant spread is 0.021 nats, except for DeepNet.

Across Blocks A and B, none of 55 runs diverged. The gradient traces suggested why: in every depth-48 run of Block B the pre-clip norm exceeds the threshold during warm-up, and PostLN's first interval reaches the largest value (1.25–2.04 against 1.13–1.16 for PreLN), so the clip was doing real work exactly where the collapse occurs. Table 16 gives Block 3 in full.

Figures 7 and 8 give the seed-level detail behind the depth table and the same numbers as a variant $\times$ depth grid. In both the depth-48 column is at a different training budget from the rest, so neither is a scaling law.

The collapse, in full. The validation loss is already ≈ 7.69 nats at the first evaluation (step 250, still inside the 500-step warm-up, at no more than half the peak rate—5×10^{-4} for the 10^{-3} runs and 1.5×10^{-3} for the 3×10^{-3} ones) and, with the single exception noted in Figure 2, never

Table 16: Block 3: depth 48, clip $= 100$, 12 k steps, three seeds. "Collapse" = final loss > 6 nats (all 12 collapsed runs here settle at 7.52–7.67; none produced a non-finite loss). $\pm$ is the across-seed s.d. to three decimals; PostLN's 0.000 at 3×10^{-4} is 0.0002 unrounded.

Variant	LR = 3×10^{-4}	LR = 1×10^{-3}	LR = 3×10^{-3}
PreLN	4.004 ± 0.005	3.779 ± 0.009	3.839 ± 0.038
PostLN	4.002 ± 0.000	*collapse* (3/3)	*collapse* (3/3)
QKNorm-PostLN	3.974 ± 0.019	*collapse* (3/3)	*collapse* (3/3)

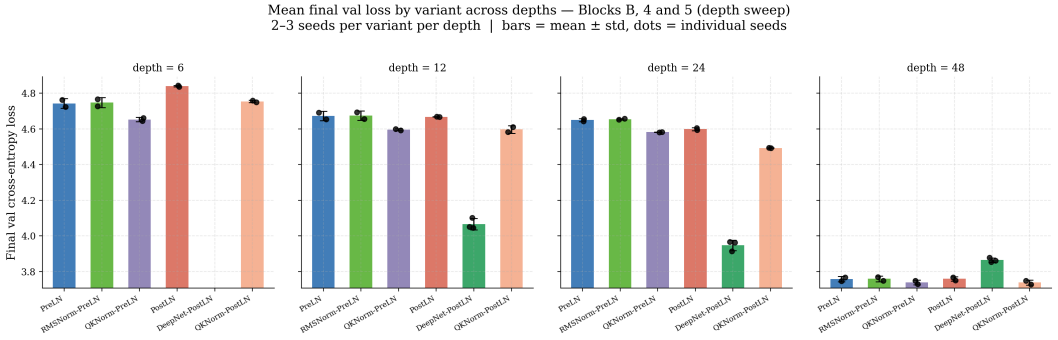


Figure 7: Final validation loss per variant, faceted by depth, with seed-level dots. Depths 6–24 are 5 k steps; depth 48 and all DeepNet cells are 20 k.

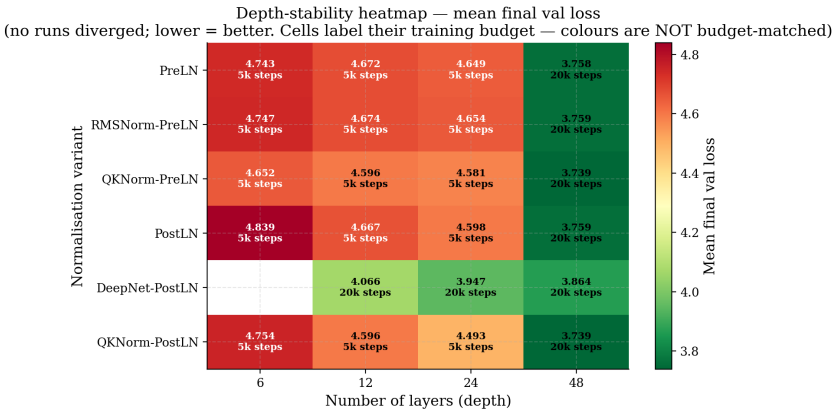


Figure 8: Mean final validation loss as a variant $\times$ depth grid, each cell labelled with its training budget. Colours compare cells at different budgets and must not be read as a scaling law. DeepNet-PostLN has no $d = 6$ cell because Block 5 was run at depths 12, 24 and 48 only.

moves; the interval-mean pre-clip gradient norm is 1.07–1.58 over the first interval for the 10^{-3} runs, against 1.11–1.12 for Pre-LN, and only 0.71–0.73 for the 3×10^{-3} runs, whose collapse is over sooner still; it then falls to 0.17–0.38 and stays there. Measuring the maximum-likelihood token entropy of the validation construction itself—the same 10,312,022 tokens over 48,286 types that `val.bin` contains—gives 7.641 nats. "Settled" means the run has stopped descending, measured from the second evaluation to the last; 1 run is excluded on that test and is the subject of Appendix B. (`scripts/measure_val_unigram_entropy.py` reproduces the entropy; it independently recovers the 20,140-window validation set reported in Section 3.) The same signature appears in the depth-12 collapses of Block 1.

F Mechanism, DeepNet and throughput

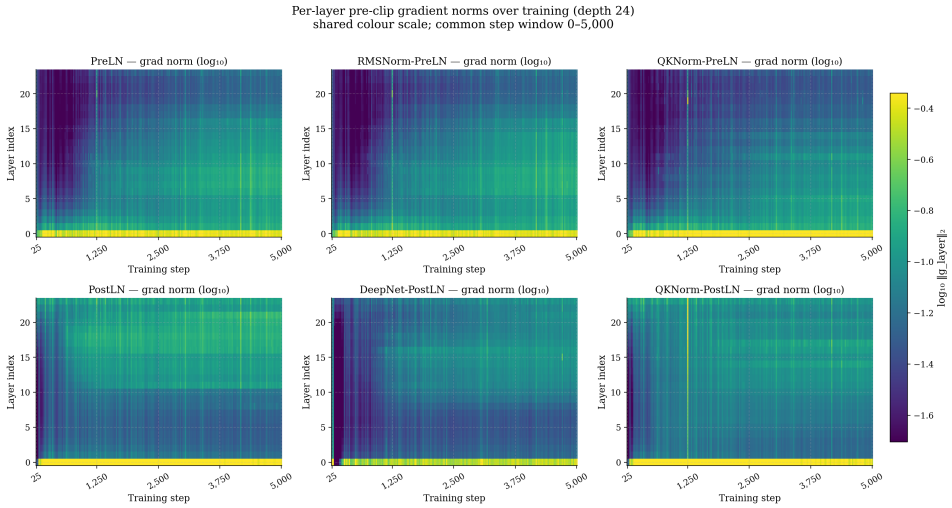


Figure 9: Per-layer pre-clip gradient ℓ_2 norms ($\log_{10}$) over training at depth 24, averaged over seeds; shared colour scale, common 0–5,000-step window, brighter is larger. The Pre-LN family is smooth and organised across depth; PostLN is uniformly higher; QK-Norm pulls the Post-LN side back toward the Pre-LN range; DeepNet is the most damped of all.

Table 17: Last-block / first-block pre-clip gradient-norm ratio at depth 24, every variant read at step 5,000. The ratio drifts upward over training, so it is only comparable at one step: DeepNet's value is 0.211 here but 0.391 at its own final step (20 k). The ± column is the across-seed sample s.d. over two or three seeds, rounded to three decimals.

Variant	last / first ratio	± sd	seeds
QKNorm-PreLN	0.114	0.022	2
PreLN	0.148	0.000	2
RMSNorm-PreLN	0.150	0.014	2
DeepNet-PostLN	0.211	0.019	3
QKNorm-PostLN	0.264	0.018	2
PostLN	0.267	0.013	2

At depth 6 the clip is engaged well past warm-up for the three variants without QK-Norm (PostLN in 78% of intervals, PreLN in 48%, RMSNorm-PreLN in 45%). From depth 12 upward PostLN is the only variant whose interval-mean pre-clip norm exceeds 1.0 after warm-up, while the largest single window grows to 2.04 at depth 48.

DeepNet. Block 5 trains DeepNet-PostLN at depths 12, 24 and 48, three seeds, 20 k steps, LR $= 3 \times 10^{-4}$ (Table 19; Figure 10). Every comparison is against 20 k-step runs at the same learning rate. DeepNet did not help in this regime: it sits 0.07 nats above vanilla PostLN at depth 12 and 0.11–0.13 above the converged band at depth 48. It *does* converge at every depth without QK-Norm

Table 18: Top: mean (max) pre-clip global gradient norm per evaluation interval, across all intervals and seeds, by variant and depth; bold marks a mean above the clip threshold of 1.0. Bottom: the percentage of intervals whose mean exceeds that threshold, which is what says whether the clip was engaged throughout training or only during warm-up. A floor of 5% at 2 seeds (2% at the 20 k horizon) corresponds to the first interval of each run and nothing else. The $d = 48$ column is given at both horizons.

Variant	$d = 6$	$d = 12$	$d = 24$	$d = 48$ (5k)	$d = 48$ (20k)
PreLN	0.95 (1.18)	0.83 (1.17)	0.72 (1.18)	0.66 (1.16)	0.74 (1.06)
RMSNorm-PreLN	0.94 (1.18)	0.83 (1.16)	0.72 (1.18)	0.66 (1.15)	0.74 (1.05)
QKNorm-PreLN	0.86 (1.12)	0.79 (1.13)	0.74 (1.15)	0.70 (1.12)	0.79 (1.04)
PostLN	**1.04 (1.23)**	0.98 (1.19)	0.92 (1.25)	0.86 (2.04)	0.85 (1.15)
QKNorm-PostLN	0.88 (1.16)	0.87 (1.21)	0.84 (1.25)	0.81 (1.33)	0.80 (1.13)
Percentage of intervals with mean pre-clip norm above the clip threshold of 1.0					
PreLN	48%	5%	5%	5%	2%
RMSNorm-PreLN	45%	5%	5%	5%	2%
QKNorm-PreLN	5%	5%	5%	5%	2%
PostLN	**78%**	40%	20%	12%	2%
QKNorm-PostLN	5%	5%	5%	5%	2%

and without any learning-rate tuning, which is its core promise. Three caveats scope the negative result tightly. Its learning rate was never swept—all runs use 3×10^{-4} for comparability—and the α/β scaling changes effective per-layer learning-rate dynamics. The regime is not the one it was designed for: Wang et al. (2024) motivate the recipe for the no-clip, very-deep setting, and at $d \leq 48$ a clip of 1.0 is already absorbing what α/β would otherwise absorb. And DeepNet's query and key projections are initialised $\approx 2.5\times$ wider than every other variant's (standard deviation ≈ 0.051 against 0.02), so the comparison to PostLN is not an α/β-only contrast; some of the depth-12 gap could be initialisation scale. Its gradient statistics, by contrast, are the best-controlled in the study; whatever the shortfall is, it is not a stability failure.

Table 19: DeepNet-PostLN against the other 20 k-step, LR $= 3 \times 10^{-4}$ cells. Depth 12 baselines are Blocks A/6; depth 48 baselines are Block 4; no 20 k baseline exists at depth 24. Bold marks the DeepNet column, the subject of the comparison, not the best value in a row.

Depth	PreLN	RMSNorm-PreLN	QKNorm-PreLN	PostLN	RMSNorm-PostLN	DeepNet-PostLN	QKNorm-PostLN
12 (Block A / 6)	3.943	3.943	3.914	3.992	3.992	**4.066**	3.941
24 (Block 5 only)	–	–	–	–	–	**3.947**	–
48 (Block 4)	3.758	3.759	3.739	3.759	–	**3.864**	3.739

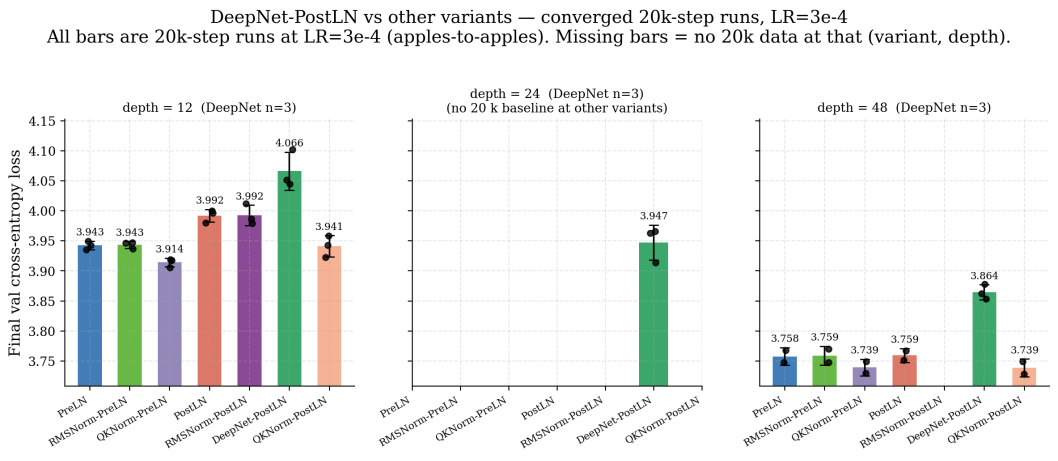


Figure 10: DeepNet-PostLN vs. the other 20 k-step variants at the same learning rate. Missing bars at $d = 24$ reflect the absence of 20 k baselines there, not zero.

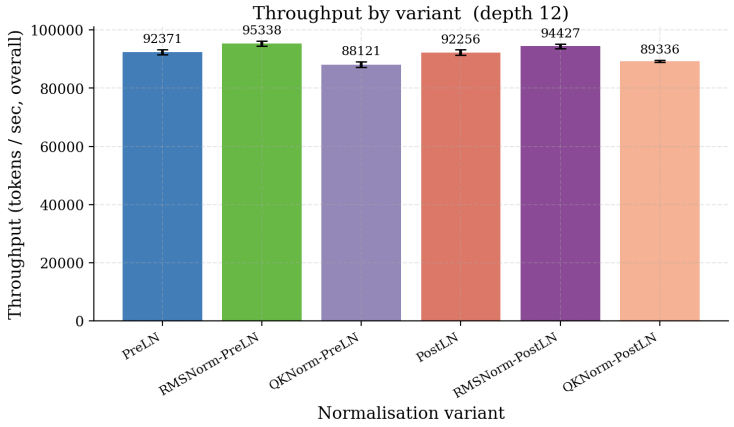


Figure 11: Overall training throughput by variant at depth 12 on H100 (mean ± s.d. over three seeds), defined as total tokens divided by total wall-clock and so inclusive of evaluation. LayerNorm → RMSNorm is +3.2% (92,371 → 95,338 tok/s).

G Full limitations and follow-ups

- **Three seeds, and what that buys.** Effect sizes and paired ranges are the evidence; p-values are not. 1 matched-block pair and the four Post-LN-penalty pairs of the selected-LR block survive Bonferroni at $\alpha = 0.01$ (Appendix C). Every directional claim in this paper holds in all three seeds; the QK-Norm claims hold under family-wise correction only at $\alpha = 0.05$.
- **Learning rates were chosen at one seed and one horizon, then applied at another.** Block 1 selects at 5 k steps; Block 2 trains for 20 k. Optimal learning rate generally falls as the horizon grows, so the Pre-LN-family picks at the top of the grid are the likeliest to be too hot at 20 k, which would inflate PreLN's measured loss and thus *understate* the selected-LR Post-LN penalty. Two of the Pre-LN picks are also within single-seed noise of the adjacent grid point. The QKNorm-PostLN pick is unambiguous.
- **No held-out test split.** The validation set selects learning rates in Block 1 and reports every headline number. Selection is over four discrete values, so the optimism is small, but the selected-LR figures are not selection-free.
- **The Post-LN closure changed mid-study.** Blocks A–5 and 8 carry a redundant final norm on Post-LN variants; Blocks 6–7 and 9–11 do not. Measured with the block effect held fixed the change is -0.0001 nats at depth 12 and LR $= 3 \times 10^{-4}$, and the headline attribution (Section 4.2) does not depend on it at all. Blocks 9 and 10 re-measured the two claims that lay outside that regime—the selected-LR margin and the depth-48 collapse—and neither moved; the depth-48 parity of Section 4.3 is the one claim still covered by extrapolation alone (Appendix B).
- **Comparisons that cross a block are noisier than those inside one.** The same configuration and seed run in Blocks A and 2 land up to 0.018 nats apart. That is not launch non-determinism—training is bit-reproducible across launches (Appendix C)—but a difference in what the two blocks read, and so an offset shared by every run in a block rather than per-run noise. This affects the RMSNorm-PostLN rows of Tables 13 and 12, the two marked rows of Table 10, and the DeepNet comparisons; every paired difference we report is taken inside one block or across a pair verified to pair.
- **The clip-isolation block is a short run.** Block 8 trains 1,000 steps, not the 12 k of Block 3. That is sufficient for the collapse/no-collapse question it was launched to answer—every collapse in this study is complete by step 250, and the two blocks share an identical learning-rate schedule over that window—but it means we report a collapse count for that cell rather than a 12 k-step loss. Its six final values do enter the plateau statistic, where the plateau is flat from step 250; they enter no other comparison.
- **Two seeds at depth 48.** A 0.021 nats spread is ≈ 2 standard errors there.

- **DeepNet is confounded** by learning rate and initialisation scale, and run outside its intended regime (Appendix F).
- **Single width, single data distribution, depth ≤ 48, no μP.** Learning-rate optima do not transfer across width without a μP-style parameterisation (Yang et al., 2021), and since every ordering here is reported at a selected learning rate, the orderings may shift at other widths; code-heavy or multilingual data could reorder the variants; DeepNet's claim is strongest beyond 100 layers. None of this is tested. The placement studies of Kim et al. (2025) and Li et al. (2025) suggest the layout comparison in particular is scale-sensitive.
- **Clip $= 100$ is not clip $= \infty$.** For the collapsing runs it is functionally infinite; for the surviving Pre-LN runs it is not quite (worst window 1.46).
- **Throughput is implementation-specific.** Attention is materialised explicitly and no kernels are fused; the 3% and 8% figures will not transfer.
- **Validation is a subsample.** Absolute values carry ≈ 0.009 nats of sampling noise; paired differences do not.

The follow-ups we would run first, in order of the uncertainty they close: (i) the depth-48 parity of Section 4.3 under the canonical closure, the one claim Blocks 9 and 10 did not reach, which would need the five depth-48 cells of Block 4 re-trained at 20 k steps rather than screened; (ii) PreLN and RMSNorm-PreLN at 10^{-3} and QKNorm-PreLN at 3×10^{-3} for 20 k steps, six runs, to test the horizon transfer; (iii) Block 8 at the full 12 k horizon, to put a 12 k-step final loss beside Block 3's; (iv) two more seeds per depth-48 cell; (v) DeepNet with GPT-2 initialisation on Q/K to separate α/β from initialisation scale.

H Compute

Table 20 accounts for all 175 runs, 64.0 B training tokens and 311 GPU-hours by block. GPU-hours are rounded per block, so the column may differ from the total by rounding.

Table 20: Compute by block, summed from the per-run artefacts. GPU-hours are training wall-clock and exclude container start-up. *GPU type absent from the committed artefacts for these runs (Blocks 3–5 were pulled without their W&B side-files and predate the provenance record); inferred from the launcher's rule that routes depth-48 runs to the larger device. Every other block records the device, either in a W&B side-file or, from Block 8 onward, in `config.json`.

Block	Study	Runs	Steps	Tokens (B)	GPU-h	GPU
A	Matched LR	15	20k	9.83	30	H100
B	Depth scan	40	5k	6.55	30	H100 / H200
1	LR sweep	20	5k	3.28	10	H100
2	Selected LR	15	20k	9.83	30	H100
3	Clip-100 stress test	27	12k	10.62	89	H200*
4	Converged $d = 48$	10	20k	6.55	53	H200*
5	DeepNet	9	20k	5.90	31	H100* / H200*
6a	RMSNorm-PostLN LR sweep	4	5k	0.66	2	H100
6b	RMSNorm-PostLN matched	3	20k	1.97	6	H100
7	Final-norm calibration	6	20k	3.93	12	H100
8	Clip isolation (screen)	6	1k	0.20	2	H200
9	Canonical closure, selected LR	6	20k	3.93	12	H100
10	Canonical closure, $d = 48$	6	1k	0.20	2	H200
11a	Determinism check (launch A)	4	2k	0.26	1	H100
11b	Determinism check (launch B)	4	2k	0.26	1	H100
Total		**175**		**64.0**	**311**	