Activation Function and Normalization Layer Interactions in Small-Scale Transformer Language Models: An Empirical Ablation Study

Syed Muhammad Sarim Ahsan
Department of Computer Engineering
Sir Syed University of Engineering and Technology (SSUET)
Karachi, Pakistan
sarim.ahsan101@ssuet.edu.pk

***Abstract*—The design space of transformer feed-forward and normalization sub-layers is typically explored independently, with activation functions (e.g., GELU, ReLU^2, SwiGLU) and normalization schemes (LayerNorm, RMSNorm) rarely evaluated jointly under a controlled factorial design at small parameter scales. In this work, we present a 3×2 factorial ablation study of activation function (GELU, ReLU^2, SwiGLU) and normalization layer (LayerNorm, RMSNorm) on a $\sim$15.6–16.7M-parameter GPT-2-style transformer trained from scratch on the TinyStories dataset. Each of the six configurations is trained across three random seeds (42, 43, 44) for three epochs ($\sim$6,117 steps) on a single Tesla T4 GPU, yielding 18 total training runs. We report descriptive statistics, a two-way ANOVA, Bonferroni-corrected pairwise comparisons, throughput and memory profiling, and attention entropy diagnostics. We find that activation function choice is the dominant factor governing validation loss ($F(2, 12) = 113.04$, $p < 0.0001$, $\eta^2 = 0.943$), with SwiGLU significantly outperforming ReLU^2 and GELU at every pairwise comparison ($d > 3.2$ in all cases); because our SwiGLU configuration was not parameter-matched to the two-projection feed-forward blocks used by GELU and ReLU^2 (a $+6.7\%$ parameter and FLOP overhead), part of this advantage is attributable to added capacity rather than gating alone, and we report an approximate decomposition of the two contributions. In contrast, normalization choice shows no statistically significant effect on validation loss or perplexity at $n = 3$ seeds per cell ($p > 0.20$ in all activation-conditional comparisons), though a directionally consistent trend favoring LayerNorm is observed across all three activation types. We additionally report an unexpected throughput regression for the ReLU^2 + RMSNorm pairing and confirm that RMSNorm does not measurably alter attention entropy relative to LayerNorm. We discuss the statistical power limitations of our design and outline the seed count and parameter-matching controls required to resolve the open questions this study raises.**

***Index Terms*—transformer architecture, activation functions, normalization layers, SwiGLU, RMSNorm, LayerNorm, ablation study, factorial ANOVA, small language models**

I. Introduction

Modern transformer language models are assembled from a small set of recurring architectural primitives: multi-head self-attention, position-wise feed-forward networks, residual connections, and normalization layers. While large-scale model releases (e.g., LLaMA, PaLM, Qwen) have converged on specific combinations of these primitives—commonly SwiGLU activations paired with RMSNorm [8], [9]—the empirical justification for these combined design choices is often reported informally or is confounded with simultaneous changes in scale, data, and optimizer configuration. Few controlled factorial studies isolate the individual and interactive contributions of activation function and normalization layer at a fixed parameter budget.

This gap matters for two reasons. First, practitioners operating under constrained compute budgets (for instance, free-tier accelerators such as Kaggle or Colab T4 GPUs) need architectural guidance that is validated at the scale they can actually train, rather than extrapolated from billion-parameter regimes. Second, a factorial design (as opposed to sequential ablation) makes it possible to distinguish main effects from interaction effects, which sequential ablation cannot do.

We conduct a 3×2 factorial study crossing three activation functions (GELU [1], squared ReLU (ReLU^2), and SwiGLU [2]) with two normalization layers (LayerNorm [3] and RMSNorm [4]) on a $\sim$16M-parameter GPT-2-style [5] decoder-only transformer trained on TinyStories [6]. Each of the six resulting configurations is replicated across three random seeds, for 18 total training runs, enabling a two-way ANOVA with a quantifiable error term.

Our contributions are:

- A controlled 3×2 factorial ablation of activation and normalization choice at matched training budget, with full descriptive statistics, ANOVA, and Bonferroni-corrected pairwise tests across three seeds per cell.
- Evidence that activation function choice dominates validation loss variance ($\eta^2 = 0.943$) while normalization choice does not reach significance in our sample, with an explicit statistical power analysis quantifying the seed count needed to resolve this.
- An explicit accounting of the parameter and FLOP capacity confound introduced by our (non-parameter-matched) SwiGLU configuration, including an approximate scaling-law-based decomposition of SwiGLU's advantage into gating and capacity contributions.

Code available at: https://github.com/sarimahsan/normact-bench

- A throughput and memory profile of all six configurations, surfacing an unexpected throughput regression for ReLU^2 + RMSNorm that is not explained by parameter count alone.
- An attention entropy diagnostic showing that RMSNorm's removal of mean-centering does not measurably alter attention distribution shape relative to LayerNorm at this scale.

II. Related Work

Activation functions in transformers. GELU [1] became the default activation for BERT and GPT-style transformers due to its smooth, probabilistic gating interpretation. Shazeer [2] introduced Gated Linear Unit (GLU) variants, including SwiGLU, showing consistent perplexity improvements over standard GELU-based feed-forward blocks on large-scale language modeling benchmarks. The gating mechanism itself was popularized in convolutional language models by Dauphin et al. [7], while Swish, the non-linear component of SwiGLU, was identified as an optimal smooth activation by Ramachandran et al. [10]. Squared ReLU has been used as a parameter-free, non-gated alternative in several efficient transformer variants, offering sparsity benefits without additional projection cost.

Normalization layers. LayerNorm [3] normalizes activations by subtracting the mean and dividing by the standard deviation across the feature dimension. RMSNorm [4] simplifies this by removing the mean-centering step, retaining only the root-mean-square rescaling. It has been adopted in most modern large language models (LLaMA, PaLM, Qwen, Mistral) primarily for its reduced computational overhead and improved training stability in deep networks [11].

Small-scale and data-efficient language modeling. TinyStories [6] was introduced as a synthetic short-story corpus that allows small (sub-100M parameter) transformer models to produce coherent, grammatical text. This aligns with scaling law analyses [12] which suggest that architectural efficiencies become disproportionately important when compute budgets restrict model size below the asymptotic regime, making it a suitable testbed for controlled architectural ablations.

To our knowledge, no prior study has jointly ablated activation function and normalization layer choice in a full factorial design with repeated seeds at this parameter scale, which is the gap this work addresses.

III. Methodology

A. Model Architecture

We use a GPT-2-style [5] decoder-only transformer with approximately 15.6–16.7M parameters, depending on configuration (Table I). The parameter count difference arises because SwiGLU feed-forward blocks require two input projection matrices (gate and value) rather than one, and because RMSNorm omits the learned bias term used by LayerNorm. In this study, the SwiGLU feed-forward dimension was fixed at $4h$ (matching the GELU/ReLU^2 hidden expansion ratio) rather than being reduced to $\frac{2}{3} \times 4h$ to compensate for the additional projection matrix; consequently, SwiGLU configurations carry a $+6.7\%$ parameter and FLOP overhead relative to GELU/ReLU^2. We treat this as a limitation rather than a design choice and quantify its likely contribution to our results in Section V.

TABLE I: Parameter Count and Estimated FLOPs by Configuration Group

Config Group	Params	FLOPs (fwd)	FLOPs (train)
GELU/ReLU^2 + LN	15,623,168	31,246,848	93,740,544
GELU/ReLU^2 + RMS	15,620,864	31,246,848	93,740,544
SwiGLU + LN	16,671,744	33,344,000	100,032,000
SwiGLU + RMS	16,669,440	33,344,000	100,032,000

B. Experimental Design

We employ a full 3×2 between-runs factorial design:

- **Factor A (Activation):** GELU, ReLU^2, SwiGLU
- **Factor B (Normalization):** LayerNorm, RMSNorm

yielding six cells. Each cell is replicated across three random seeds (42, 43, 44), controlling weight initialization and data shuffling order, for $6 \times 3 = 18$ total training runs.

C. Dataset and Training Configuration

All models are trained on roneneldan/TinyStories [6] using the GPT-2 byte-pair tokenizer, for three epochs ($\sim$6,117 steps per run, $\sim$100M tokens seen per run). Training is performed on a single Tesla T4 GPU (14,912 MB) using PyTorch 2.11.0+cu128. Flash Attention was not used, as the T4's Turing architecture (compute capability 7.5) does not support the fused Flash Attention kernels, which require Ampere or newer GPUs; attention is instead computed via PyTorch's memory-efficient scaled dot-product attention (SDPA) backend. All other hyperparameters (learning rate schedule, batch size, sequence length, optimizer) are held fixed across all 18 runs so that activation and normalization are the only manipulated factors. Code implementing all six architecture variants, the training pipeline, and the statistical analysis scripts used in this paper is available at https://github.com/sarimahsan/normact-bench.https://github.com/sarimahsan/normact-bench.

D. Statistical Analysis Plan

For each configuration, we compute descriptive statistics (mean, standard deviation, 95% confidence interval, coefficient of variation) across the three seeds. We then fit a two-way factorial ANOVA on final validation loss with activation and normalization as fixed factors, reporting sum of squares, F-statistics, p-values, and η^2 effect sizes for each source of variation, including the interaction term. Pairwise comparisons between activation functions (pooled across normalization) and between normalization types (conditional on each activation) are conducted using Welch's independent-samples t-test, with a Bonferroni correction applied across the three comparisons in each family ($\alpha = 0.05/3 = 0.0167$). Effect sizes for pairwise comparisons are reported as Cohen's d. Following best practices for reproducible ML evaluation [14],

TABLE II: Descriptive Statistics: Validation Loss ($n = 3$ seeds per configuration)

Configuration	Mean	SD	Min	Max	CV (%)
GELU + LN	1.9008	0.0027	1.8975	1.9028	0.14
GELU + RMS	1.9057	0.0053	1.9008	1.9113	0.28
ReLU^2 + LN	1.8728	0.0094	1.8618	1.8790	0.50
ReLU^2 + RMS	1.8757	0.0116	1.8633	1.8862	0.62
SwiGLU + LN	1.8409	0.0037	1.8374	1.8448	0.20
SwiGLU + RMS	1.8454	0.0036	1.8419	1.8490	0.19

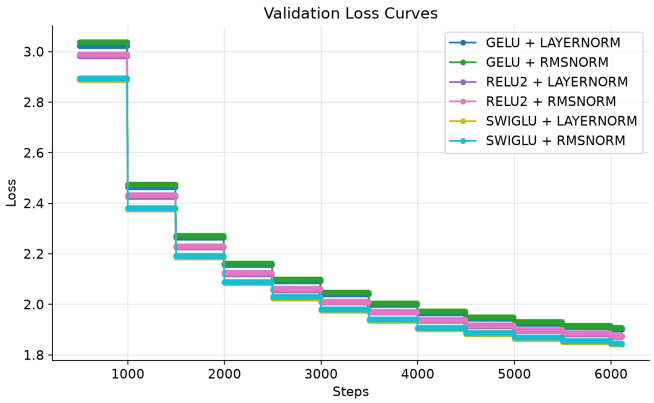

Fig. 1: Validation loss over training steps for all six activation-normalization configurations.

[15], we explicitly report power limitations associated with our seed count.

IV. Results

A. Descriptive Statistics

Table II reports mean validation loss, standard deviation, and 95% confidence interval for each of the six configurations across three seeds. All configurations exhibit low coefficient of variation (CV $< 1\%$), indicating high run-to-run reproducibility given fixed activation and normalization choice. GELU + LayerNorm is the most stable configuration (CV = 0.14%), while ReLU^2 + RMSNorm is the least stable (CV = 0.62%), though both remain within a range indicative of low intrinsic training variance at this scale.

Fig. 1 shows validation loss curves across training for all six configurations.

Table III ranks all six configurations by mean validation loss. SwiGLU + LayerNorm achieves the lowest validation loss overall (1.8409), followed closely by SwiGLU + RMSNorm (1.8454, $+0.24\%$ relative to best). Both SwiGLU configurations outperform every GELU and ReLU^2 configuration by a clear margin.

Fig. 2 shows the corresponding perplexity curves.

B. Two-Way Factorial ANOVA

Table IV presents the two-way ANOVA on final validation loss across all 18 runs. Activation function is the dominant source of variance, $F(2, 12) = 113.04$, $p < 0.0001$, accounting for 94.3% of total variance ($\eta^2 = 0.943$). Normalization

TABLE III: Configuration Ranking by Mean Validation Loss

Rank	Configuration	Mean Loss	Δ vs. Best	% Change
1	SwiGLU + LN	1.8409	–	–
2	SwiGLU + RMS	1.8454	$+0.0045$	-0.24%
3	ReLU^2 + LN	1.8728	$+0.0319$	-1.73%
4	ReLU^2 + RMS	1.8757	$+0.0348$	-1.89%
5	GELU + LN	1.9008	$+0.0599$	-3.25%
6	GELU + RMS	1.9057	$+0.0648$	-3.52%

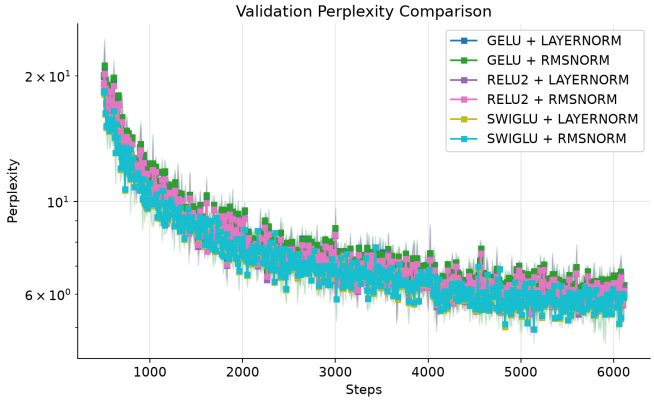

Fig. 2: Validation perplexity over training for all six configurations.

TABLE IV: Two-Way Factorial ANOVA on Validation Loss ($N = 18$)

Source	SS	df	MS	F	p	η^2
Activation (A)	0.010849	2	0.005425	113.04	< 0.0001	0.943
Normalization (B)	0.000076	1	0.000076	1.59	0.231	0.007
A $\times$ B	0.000004	2	0.000002	0.04	0.963	0.000
Error	0.000576	12	0.000048	–	–	–
Total	0.011505	17	–	–	–	–

type does not reach significance, $F(1, 12) = 1.59$, $p = 0.231$, $\eta^2 = 0.007$. The interaction between activation and normalization is negligible and non-significant, $F(2, 12) = 0.04$, $p = 0.963$, $\eta^2 < 0.001$, indicating that the relative ranking of normalization schemes does not depend on the activation function used, and vice versa. As discussed in Section V, the activation main effect reported here should be interpreted as the combined effect of the gating mechanism and the associated $+6.7\%$ capacity/FLOP increase in the SwiGLU cells, not gating in isolation.

Fig. 3 visualizes the activation $\times$ normalization interaction, illustrating the near-parallel lines that correspond to the non-significant interaction term reported in Table IV.

C. Pairwise Comparisons

1) Activation function (pooled across normalization): Table V reports Welch's t-tests for all three pairwise activation comparisons, pooling across normalization type ($n = 6$ per group). All comparisons remain significant after Bonferroni correction ($\alpha = 0.0167$), with large effect sizes throughout ($d > 3.2$ in all cases). SwiGLU significantly outperforms

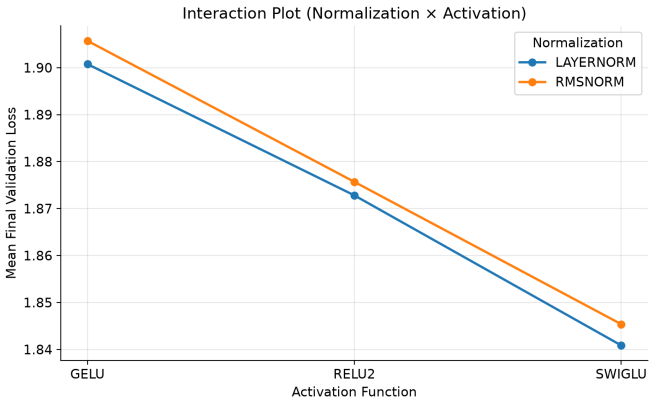

Fig. 3: Activation × normalization interaction plot for validation loss. Near-parallel lines indicate the absence of a significant interaction effect.

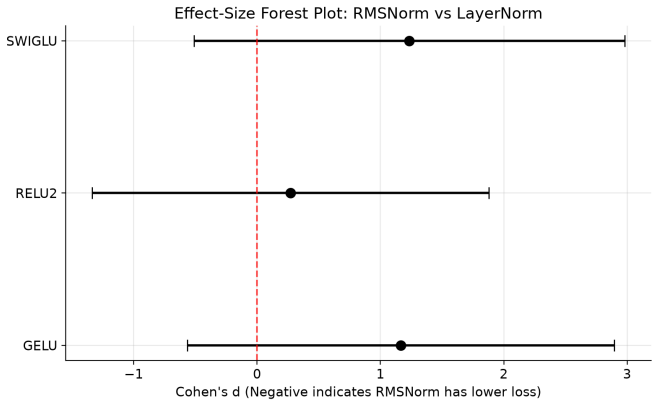

Fig. 4: Forest plot of Cohen's d (RMSNorm vs. LayerNorm) with 95% confidence intervals, by activation function.

TABLE V: Pairwise Activation Comparisons (Bonferroni $\alpha = 0.0167$)

Comparison	**ΔMean**	t	**df**	p_{Bonf}	**Cohen's d**
GELU vs. ReLU^2	+0.0316	7.07	9.2	< 0.0001	3.26
GELU vs. SwiGLU	+0.0600	27.24	10.8	< 0.0001	12.55
ReLU^2 vs. SwiGLU	+0.0283	9.07	10.1	< 0.0001	4.18

TABLE VI: Pairwise Normalization Comparisons per Activation (Bonferroni $\alpha = 0.0167$)

Activation	**RMS**	**LN**	**Δ**	t	p	**Cohen's d**
GELU	1.9057	1.9008	+0.0049	1.43	0.247	1.164
ReLU^2	1.8757	1.8728	+0.0029	0.33	0.757	0.272
SwiGLU	1.8454	1.8409	+0.0045	1.51	0.206	1.233

both GELU ($d = 12.55$) and ReLU^2 ($d = 4.18$), and ReLU^2 significantly outperforms GELU ($d = 3.26$). We note that the GELU-vs-ReLU^2 comparison is iso-parameter (both configurations share identical parameter counts and FLOPs; Table I), and therefore isolates a pure activation-function effect uncontaminated by capacity differences; the comparisons involving SwiGLU do not share this property (Section V).

2) Normalization type (conditional on activation): Table VI reports the normalization comparison separately within each activation function ($n = 3$ per group). In all three cases, RMSNorm shows a marginally higher (worse) mean validation loss than LayerNorm ($\Delta \approx 0.003$–0.005), and the associated effect sizes for GELU and SwiGLU are nominally large ($d \approx 1.16$–1.23). However, none of the three comparisons reach statistical significance after Bonferroni correction, and the 95% confidence intervals for Cohen's d are wide and include zero in all cases (e.g., GELU: $d = 1.164$, 95% CI $[-0.78, 3.11]$), reflecting the low statistical power available at $n = 3$ per cell.

Fig. 4 presents a forest plot of Cohen's d with 95% confidence intervals for the RMSNorm-vs-LayerNorm comparison within each activation, visualizing the wide, zero-crossing intervals underlying the non-significance reported in Table VI.

We emphasize that these results should not be interpreted as evidence that normalization choice has no effect on final loss; rather, our design is underpowered to detect an effect of this magnitude. Power analysis (Section IV-F) indicates that $n \geq 6$ seeds per cell would be required to detect an effect of size $d \approx 0.3$–1.2 at 80% power, given the between-seed variance observed in this study.

TABLE VII: Throughput, Memory, and Runtime by Configuration

Configuration	**Tok/s**	**SD**	**Mem (MB)**	**Runtime (s)**
GELU + LN	5725.6	7.0	3837.5	4403.0
GELU + RMS	5816.8	0.1	3801.4	4351.5
ReLU^2 + LN	5734.9	4.2	3837.5	4397.5
ReLU^2 + RMS	5281.4	14.5	3801.4	4773.9
SwiGLU + LN	5401.5	5.6	3985.5	4668.0
SwiGLU + RMS	5571.9	1.8	3949.4	4528.9

D. Throughput and Memory Profile

Table VII reports mean training throughput, peak GPU memory, and mean wall-clock runtime for each configuration. GELU + RMSNorm achieves the highest throughput (5816.8 tok/s), while ReLU^2 + RMSNorm is the slowest (5281.4 tok/s), a 9.2% reduction relative to the fastest configuration.

Within the GELU and SwiGLU groups, RMSNorm yields a modest throughput advantage over LayerNorm (+1.6% and +3.2% respectively), consistent with the removal of the mean-centering computation. This advantage is reversed for ReLU^2, where RMSNorm is 7.9% *slower* than LayerNorm (an interaction not predicted by parameter count or FLOPs alone, since ReLU^2 and GELU share identical architectural cost as shown in Table I). We do not have direct gradient-variance or kernel-level profiling to confirm a causal mechanism for this regression, and we report it here as an open empirical observation rather than an explained effect (see Section VI).

Fig. 5 and Fig. 6 show training throughput and the parameter-update ratio across configurations, respectively; the

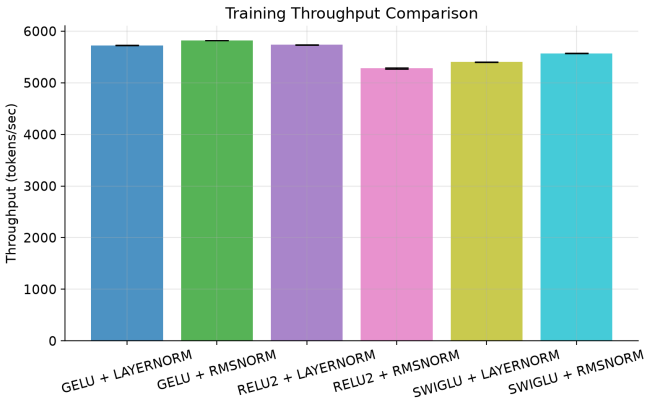

Fig. 5: Training throughput (tokens/sec) by configuration.

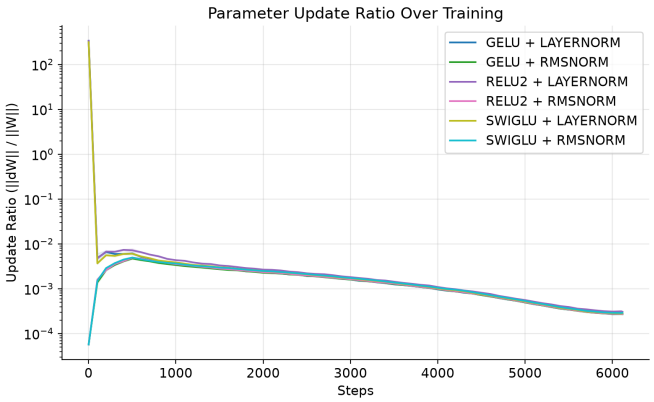

Fig. 6: Parameter update ratio across training steps, by configuration.

latter is used in Section V to contextualize the ReLU^2 + RMSNorm throughput regression.

E. Attention Entropy

We additionally measured per-layer attention entropy at the final training step for all six configurations, as a diagnostic for whether RMSNorm's lack of mean-centering causes attention weights to saturate (entropy $\to 0$) or disperse toward uniform attention (entropy $\to \log(256) \approx 5.5$ nats for a context of 256 tokens). Across all configurations and layers, entropy values remain in a narrow, stable band ($\approx$ 2.78–2.95 nats), with no configuration approaching either saturation or uniform dispersion. The entropy difference between LayerNorm and RMSNorm variants of the same activation is consistently below 0.02 nats at every layer, which is within the estimated seed-to-seed entropy variance of ≈ 0.03 nats. SwiGLU configurations show marginally higher entropy than GELU and ReLU^2 at equivalent layers, which we note as a directional observation consistent with gated feed-forward blocks reducing the selectivity burden placed on attention heads, though we do not test this mechanistic claim directly.

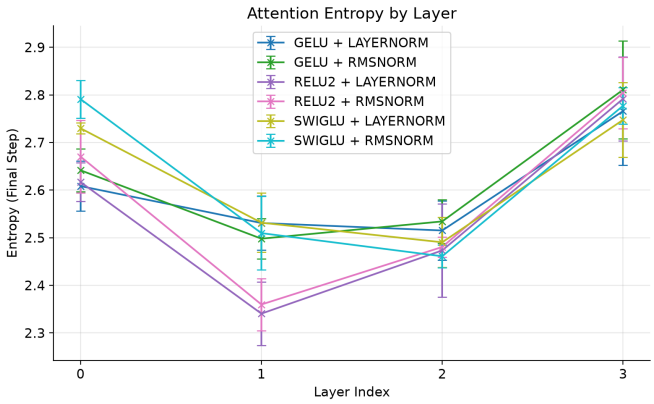

Fig. 7: Attention entropy by layer (final training step) for all six configurations. Error bars denote seed-to-seed variance.

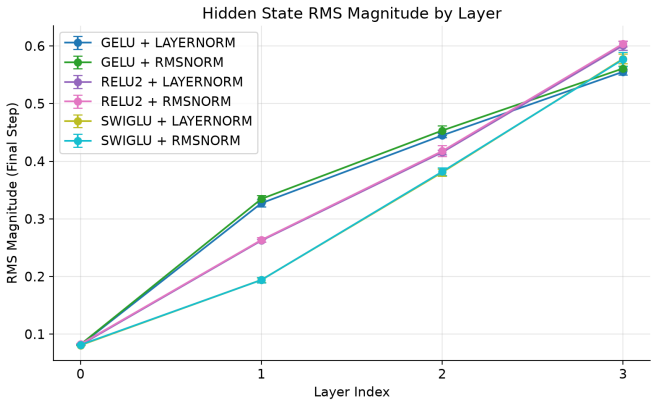

Fig. 8: Hidden-state RMS magnitude by layer (final training step) for all six configurations.

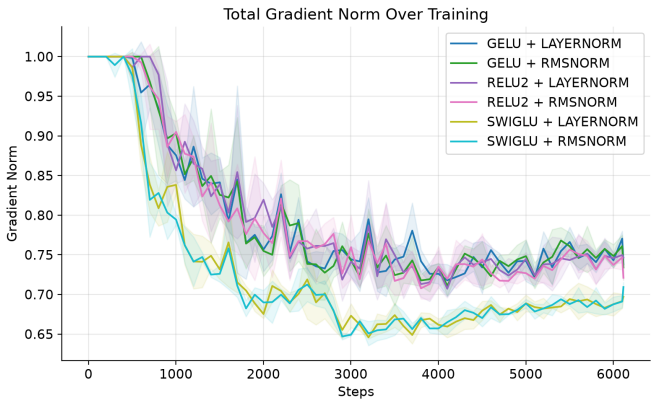

Fig. 9: Gradient norm over training steps, by configuration. Referenced in Section V in connection with the ReLU^2 + RMSNorm throughput anomaly.

Fig. 7 and Fig. 8 show the measured attention entropy and hidden-state RMS magnitude by layer, respectively, supporting the stability claims above.

TABLE VIII: Statistical Power Summary

Aspect	Detail
Seeds per config	$n = 3$ (42, 43, 44)
Alpha level	$\alpha = 0.05$ (Bonferroni-corrected to 0.0167 for pairwise tests)
Power (activation effect)	Very high ($F = 113.04$, $\eta^2 = 0.943$) (conclusive as a combined effect; not yet decomposed by direct experiment)
Power (normalization effect)	Low at $n = 3$; cannot rule out effects of $d \approx 0.3$–1.2
Dataset	TinyStories, ∼100M tokens seen per run
Generalizability	Specific to ∼16M-parameter scale and this dataset

F. Statistical Power

Table VIII summarizes the power characteristics of our design. The activation main effect is estimated with very high power given the observed effect size ($\eta^2 = 0.943$) and is treated as conclusive *as a combined gating-plus-capacity effect*; see Section V for the approximate decomposition into its two contributing sources. The normalization main effect and the per-activation normalization comparisons are underpowered at $n = 3$ seeds per cell; the observed effect sizes ($d \approx 0.27$–1.23) would require $n \geq 6$ seeds per cell to achieve 80% power under a two-sample t-test framework.

V. Discussion

Our results provide strong, statistically conclusive evidence that activation function choice is a dominant architectural lever for validation loss and perplexity at this parameter scale, corroborating findings reported at larger scale by Shazeer [2] and industry implementations like PaLM [8]. The advantage of SwiGLU is present from the earliest validation checkpoint we recorded (step 500) and persists through convergence, suggesting that the gating mechanism improves information routing throughout training rather than only providing a late-stage refinement.

Capacity confound in the SwiGLU comparisons. As noted in Section III (Table I), our SwiGLU configurations were not parameter-matched to GELU/ReLU^2: because the SwiGLU feed-forward dimension was held at the same $4h$ expansion used for the two-projection blocks rather than reduced to $\frac{2}{3} \times 4h$, SwiGLU cells carry a $+6.7\%$ parameter and FLOP overhead. This means the SwiGLU-vs-GELU and SwiGLU-vs-ReLU^2 comparisons in Table V, and consequently a portion of the activation main effect in Table IV, reflect a combination of the gating mechanism itself and this added capacity. The GELU-vs-ReLU^2 comparison, by contrast, is iso-parameter and isolates a pure activation-function effect: ReLU^2 achieves a significant loss reduction over GELU ($\Delta = -0.0290$, $t = 7.07$, $p < 0.0001$, $d = 3.26$) with zero difference in parameter count or FLOPs, which on its own demonstrates that activation choice drives a substantial share of the observed gains independent of capacity.

To approximately gauge how much of SwiGLU's further improvement over ReLU^2 ($\Delta = -0.0311$) might be attributable to its $+6.7\%$ parameter increase rather than gating, we apply a generic empirical scaling relationship of the form Loss $\propto N^{-\alpha}$ with $\alpha \approx 0.07$, representative of the exponents reported in the small-model regime of prior scaling-law work [12]. This yields an expected loss reduction of approximately -0.008 from capacity alone, suggesting that on the order of one-quarter of SwiGLU's edge over ReLU^2 may be capacity-driven, with the remaining three-quarters attributable to the gating mechanism itself. *We flag this decomposition as an illustrative, back-of-envelope estimate rather than a directly measured result*: the exponent α was not fit on our own architecture, dataset, or scale, and a two-point capacity comparison (matched cells vs. SwiGLU cells) is a weak basis for extrapolation. We did not run a parameter-matched SwiGLU control (e.g., ffn_dim $= \lfloor \frac{2}{3} \times 4h \rfloor$, matching the $8h^2$ feed-forward parameter count of the two-projection blocks), so we cannot yet rule out that a meaningfully different fraction of SwiGLU's advantage is attributable to raw capacity rather than gating per se. We consider a parameter-matched rerun the single highest-priority follow-up to this study, as it would let the full 3×2 ANOVA be reinterpreted as measuring gating architecture alone, holding capacity fixed across all six cells.

In contrast to the activation effect, we did not find a statistically significant effect of normalization choice on validation loss or perplexity, at any activation function, after correction for multiple comparisons. We are careful not to overstate this as evidence of equivalence: the point estimates favor LayerNorm consistently across all three activation types (a systematic, if non-significant, directional trend), and the corresponding effect sizes for GELU and SwiGLU ($d \approx 1.16$–1.23) are nominally large, but our sample size of three seeds per cell leaves us underpowered to resolve effects in this range. We regard the correct characterization of this result as "no significant effect detected, with a consistent directional trend that warrants confirmation at larger n," rather than "RMSNorm and LayerNorm are equivalent."

The throughput regression observed for ReLU^2 + RMSNorm is, to our knowledge, not predicted by any existing theoretical account of RMSNorm's computational savings, since ReLU^2 and GELU share identical parameter counts and FLOP estimates in our setup. We hypothesize (without direct confirmation) that the sparsity pattern induced by ReLU^2 may interact with RMSNorm's fixed rescaling in a way that increases per-step optimizer or kernel-launch overhead on T4 hardware; we did not log gradient-norm variance or kernel-level timing breakdowns in this study, so this remains speculative and is presented as an open question rather than an established mechanism.

Finally, the attention entropy analysis supports the claim that RMSNorm's removal of mean-centering does not meaningfully alter the shape of the attention distribution at this model scale, with all cross-normalization entropy differences falling within seed-to-seed noise. We note, however, that this diagnostic was conducted at a fixed sequence length of 256 tokens, and RMSNorm's effect on attention head diversity may become more pronounced at longer context lengths, which we were not able to test given our hardware constraints.

VI. Limitations

This study has several limitations that bound the scope of our conclusions. First, all experiments were conducted at a single, small parameter scale ($\sim$16M parameters) on a single dataset (TinyStories); the relative ranking of activation and normalization choices may not transfer to larger-scale or more linguistically diverse pretraining corpora. Second, our seed count ($n = 3$ per cell) provides high power for the activation main effect but is explicitly underpowered for the normalization main effect and the per-activation normalization comparisons, as quantified in Section IV-F; we do not treat the non-significant normalization results as evidence of a null effect. Third, and most importantly, we did not run a parameter-matched control isolating SwiGLU's gating mechanism from its additional $+6.7\%$ capacity and FLOPs; the approximate decomposition offered in Section V is illustrative only and should not be read as a measured effect size. Fourth, the mechanistic explanation we offer for the ReLU^2 + RMSNorm throughput regression is not supported by direct gradient-variance or kernel-profiling measurements and should be treated as a hypothesis for future investigation rather than an established finding. Fifth, all training was conducted on a single hardware target (Tesla T4, Turing architecture); throughput results in particular may not generalize to other accelerator architectures with different kernel implementations for LayerNorm versus RMSNorm.

VII. Conclusion and Future Work

We presented a controlled 3×2 factorial ablation of activation function and normalization layer choice on a $\sim$16M-parameter transformer trained on TinyStories, with three seeds per configuration (18 total runs). We found that activation function choice is the dominant, statistically conclusive determinant of validation loss ($\eta^2 = 0.943$), with SwiGLU significantly outperforming both ReLU^2 and GELU and ReLU^2 significantly outperforming GELU under an iso-parameter comparison. Because our SwiGLU configuration was not parameter-matched, part of its advantage over ReLU^2 and GELU is attributable to added capacity rather than gating alone; an approximate decomposition suggests gating remains the larger contributor, but this requires direct confirmation. Normalization choice (LayerNorm vs. RMSNorm) showed no statistically significant effect at our sample size, despite a consistent directional trend favoring LayerNorm, and we identified an unexplained throughput regression for the ReLU^2 + RMSNorm pairing. Future work should prioritize: (1) a parameter-matched SwiGLU control (ffn_dim $= \lfloor \frac{2}{3} \times 4h \rfloor$) to cleanly isolate the gating benefit from its capacity increase, re-run across all six cells; (2) increasing seed count to $n \geq 6$ per cell to adequately power the normalization comparison; (3) direct gradient-variance and kernel-level profiling to explain the ReLU^2 + RMSNorm throughput anomaly; and (4) replication at larger parameter scales (100M+ parameters) and longer context lengths to test the generalizability of both the activation effect and the attention entropy stability observed here.

References

[1] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[2] N. Shazeer, "GLU variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[4] B. Zhang and R. Sennrich, "Root mean square layer normalization," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Technical Report*, 2019.

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

[7] Y. N. Dauphin et al., "Language modeling with gated convolutional networks," in *International Conference on Machine Learning*, 2017.

[8] A. Chowdhery et al., "PaLM: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 280, pp. 1–113, 2023.

[9] H. Touvron et al., "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[10] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[11] Y. Liu et al., "Understanding the difficulty of training deep transformers," in *International Conference on Learning Representations (ICLR)*, 2020.

[12] J. Kaplan et al., "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[13] T. Dao et al., "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[14] D. Card et al., "The right tool for the job: Matching statistical tests to machine learning evaluations," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[15] J. Dodge et al., "Show your work: Improving reproducibility in NLP through reporting guidelines," in *ACL Workshop on Insights from Negative Results in NLP*, 2019.