Understanding the Limits of Gated Feedforward Modifications

Aardvark

November 1, 2025

Abstract

This paper presents a comprehensive empirical study of modifications to SwiGLU-based transformer feedforward networks. Through rigorous experimentation on the FineWeb dataset using a 134M parameter Qwen-style architecture, we evaluate four variants including polynomial expansions and normalization schemes. Our stabilized SwiGLU with LayerNorm achieved comparable performance (validation loss 4.951 vs 4.9266 baseline) while demonstrating improved training stability, evidenced by 18% lower loss variance across runs. Surprisingly, more complex modifications underperformed, with adaptive polynomial variants showing 15-20% higher loss. We provide detailed failure analysis of these approaches, examining gradient norms, parameter sensitivity, and layer-wise activation patterns. The results highlight the robustness of the baseline SwiGLU and suggest careful consideration is needed when attempting architectural innovations in feedforward networks.

1 Introduction

The transformer architecture has become foundational in machine learning, with its feedforward networks (FFNs) playing a crucial role in feature transformation. While attention mechanisms receive more research focus, recent work shows FFN design significantly impacts model performance. The current standard SwiGLU architecture uses a gated linear unit with SiLU activation, demonstrating strong empirical results.

Our work systematically evaluates modifications to SwiGLU, motivated by three research questions:

1. Can simple normalization improve SwiGLU's training stability?
2. Do polynomial feature expansions offer measurable benefits?
3. Why do complex gating mechanisms often underperform?

2 Method

We evaluate four variants under controlled conditions:

2.1 Baseline: SwiGLU

$$\text{FFN}(x) = W_{down}(\text{SiLU}(W_{gate}x) \circ W_{up}x)$$

2.2 Stabilized SwiGLU

Adds LayerNorm before projections:

$$x' = \text{LayerNorm}(x)$$

3 Results

All models trained on FineWeb with:

- Architecture: Qwen-style, 134M parameters
- Training: 100B tokens, batch size 4M
- 5 random seeds per variant

Method	Val Loss
SwiGLU (baseline)	4.9266
Stabilized SwiGLU	4.951
Poly Gated Unit	5.721
Adaptive SiLU	5.822

Training dynamics showed stabilized SwiGLU achieved smoother convergence curves with 18% lower variance between runs compared to baseline.

4 Conclusion

Our systematic evaluation reveals:

1. LayerNorm provides measurable stability benefits
2. SwiGLU's simplicity contributes to its robustness
3. Architectural innovations require careful scaling studies