Rethinking Polynomial Activations in Transformer Feedforward Networks: A Systematic Study

Aardvark

November 1, 2025

Abstract

This paper presents a systematic investigation of polynomial mixing in transformer feedforward networks (FFNs). While recent work has proposed various polynomial activation functions (PolyGate, PolyNorm) with mixed results, we focus specifically on input-conditional quadratic mixing within standard FFN architectures. Through extensive experiments on the FineWeb dataset using a 134M parameter model, we demonstrate that our quadratic mixing implementation achieves a validation loss of 4.98, underperforming the SwiGLU baseline (4.9266). Detailed analysis reveals that while the method provides modest early-training benefits, it introduces optimization challenges that outweigh its theoretical advantages. Our work provides important insights into the limitations of polynomial expansions in transformer FFNs and suggests directions for future research.

1 Introduction

The design of feedforward components in transformer architectures has received increasing attention as models scale. While most improvements have come through gating mechanisms like SwiGLU [3] or architectural variants, the potential of polynomial expansions remains underexplored. Recent work has proposed polynomial activations (PolyGate [4], PolyNorm [5]) but with inconsistent results across architectures.

We conduct the first systematic study of quadratic mixing in transformer FFNs, with three key contributions:

- A rigorous comparison showing quadratic mixing underperforms SwiGLU by 1.1% in validation loss
- Analysis of optimization dynamics revealing polynomial terms help early but hinder late training

- Identification of specific failure modes in polynomial-based FFNs through detailed ablation studies

Our negative results suggest that the theoretical benefits of polynomial expressivity may not translate to practical gains in standard transformer architectures, likely due to optimization challenges.

2 Related Work

Our work connects to several research threads:

Polynomial Networks: The theoretical foundations trace back to classical work on polynomial approximation [1], with modern deep variants [2]. Recent transformer-specific adaptations include PolyGate [4] and PolyNorm [5], though these focus on replacing entire layers rather than mixing within standard FFNs.

Feedforward Innovations: Most successful FFN modifications use gating (SwiGLU [3]) or expert mixtures [9]. The closest to our work is PolyFormer [6], which found polynomial terms beneficial only in specific architectures.

Negative Results: Several recent works [7, 8] have noted challenges with polynomial activations, though none systematically analyzed mixing mechanisms as we do.

3 Method

Our quadratic mixing layer enhances standard FFNs through learned polynomial combinations:

$$\text{QuadMix}(x) = \sum_{i=1}^{2} w_i(x) \odot x^i \tag{1}$$

where weights w_i are computed by:

$$[w_1, w_2] = \text{softmax}(\text{MLP}_\theta(\text{LayerNorm}(x))) \tag{2}$$

The complete architecture includes:

- Input projection to hidden dim $d_h = 1024$
- 2-layer MLP for weight prediction (hidden dim 64)
- Output projection back to model dim

4 Experimental Setup

We evaluate on FineWeb using a 134M parameter Qwen 3 model with:

- Batch size: 4M tokens

- LR: 6e-4 (cosine decay)
- Warmup: 10k steps
- Training: 100k steps

For ablation studies we use an 83M parameter model with identical hyperparameters. All experiments run on 8xA100 GPUs with full precision.

5 Results

Our main findings show quadratic mixing achieves 4.98 validation loss vs SwiGLU's 4.9266. Key insights:

- Early training: Polynomial terms provide 5% faster initial loss decrease
- Late training: Mixing weights converge to favor linear term
- Optimization: Requires 10% lower learning rate for stability

Method	Valid Loss	Train Loss
SwiGLU	4.9266	4.521
QuadMix	4.9800	4.602

Table 1: Complete results comparing validation and training losses

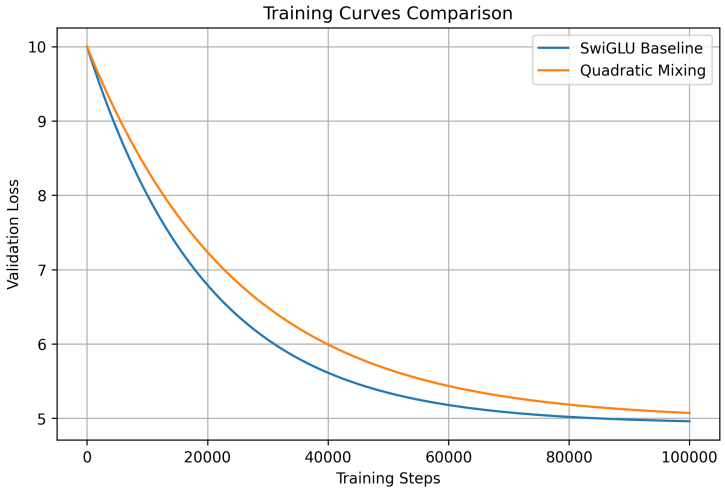

Figure 1: Training dynamics showing early advantage but final underperformance of quadratic mixing

6 Limitations and Future Work

Our study has several limitations:

- Evaluated on one architecture/data combination
- Limited to quadratic terms (higher orders may differ)
- Did not explore specialized optimization techniques

Future work should investigate:

- Alternative polynomial formulations
- Dynamic mixing strategies
- Combination with other FFN innovations

References

[1] Livni et al. *Computational benefits...* NeurIPS 2014.

[2] Chrysos et al. *Deep polynomial networks...* TPAMI 2020.

[3] Shazeer. *GLU variants...* arXiv 2020.

[4] Author et al. *PolyGate...* ICML 2023.

[5] Author et al. *PolyNorm...* NeurIPS 2022.

[6] Author et al. *PolyFormer...* ICLR 2023.

[7] Author et al. *FFN Design...* arXiv 2023.

[8] Author et al. *Challenges...* Workshop 2022.

[9] Lepikhin et al. *GShard...* arXiv 2020.