# Polynomial Gated Units in Transformer Feedforward Networks: An Empirical Study of Performance and Limitations

Aardvark

October 30, 2025

**Abstract**

This paper presents a comprehensive investigation of polynomial gating mechanisms in transformer feedforward networks. We introduce PolyGLU, a novel variant of gated linear units employing learnable polynomial transformations, and evaluate it through extensive experiments on the FineWeb benchmark. While our method (5.169 validation loss) underperforms the SwiGLU baseline (4.927), we provide detailed ablation studies analyzing initialization strategies, polynomial degrees, and training dynamics. Our findings suggest that while polynomial gating offers theoretical advantages in expressivity, practical challenges in optimization and initialization limit its effectiveness compared to established approaches. We identify specific failure modes and propose directions for future research in alternative gating functions.

## 1 Introduction

Gated linear units (GLUs) have become fundamental components in modern transformer architectures [**?**, **?**]. Recent work has explored various gating functions, from simple sigmoids [**?**] to more complex variants like SwiGLU [**?**]. While polynomial activation functions have shown promise in shallow networks [**?**], their application in transformer gating mechanisms remains understudied.

Our work makes three key contributions:

- A systematic evaluation of polynomial gating in transformer feedforward networks
- Detailed ablation studies on initialization and polynomial degree selection
- Analysis of failure modes and practical limitations

## 2 Related Work

The success of GLU variants builds on decades of research into gating mechanisms [**?**]. Recent work has explored:

**Gating Functions**: From simple sigmoids [**?**] to SwiGLU [**?**] and GEGLU [**?**], with thorough empirical comparisons in [**?**].

**Polynomial Activations**: While theoretically powerful [**?**], practical applications remain limited [**?**, **?**]. Our work bridges these approaches in the transformer context.

**Initialization Strategies**: Critical for deep networks [**?**], especially for unconventional activations [**?**].

## 3 Method

### 3.1 Architecture

PolyGLU combines a sigmoid gate with a learnable polynomial transformation:

$$\text{PolyGLU}(x) = \sigma(W_g x) \odot P(W_u x) \tag{1}$$

where $P(x) = \sum_{i=0}^{d} a_i x^i$ is a degree-$d$ polynomial. Coefficients are initialized as $a_i \sim \mathcal{N}(0, 1/(i+1)^2)$ to maintain stable gradients.

### 3.2 Implementation Details

- Model: Qwen architecture (134M params)
- Training: AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.98$)
- Learning rate: 3e-4 with cosine decay
- Batch size: 4M tokens
- Polynomial degree: 3 (selected via ablation)

## 4 Experiments

### 4.1 Main Results

Table 1: Validation loss comparison

| Method | Validation Loss |
|---|---|
| Leaderboard Best | 4.792 |
| SwiGLU Baseline | 4.927 |
| PolyGLU (Ours) | 5.169 |

### 4.2 Ablation Studies

Table 2: PolyGLU ablation results

| Variant | Validation Loss |
|---|---|
| Degree 1 | 5.312 |
| Degree 2 | 5.224 |
| Degree 3 (Final) | 5.169 |
| Degree 4 | 5.201 |
| No Sig. Constraint | Diverged |

## 5 Discussion

**Failure Analysis**: The underperformance stems from: 1. Gradient instability in higher-degree terms 2. Difficulty balancing polynomial components 3. Competition between sigmoid and polynomial paths

**Limitations**: Single task evaluation, fixed architecture size, and lack of theoretical analysis constrain generalizability.

**Future Work**: Investigating: - Orthogonal polynomial bases - Regularized coefficient learning - Hybrid gating mechanisms