Normalization Layers

LayerNorm & RMSNorm

Comprehensive Demo with Visualizations

Seed: 42

1. LayerNorm: normalizes to zero mean, unit variance (GPT-2, BERT)
2. RMSNorm: rescales by root-mean-square, no mean subtraction (LLaMA, Mistral)
3. Both behave identically at training and inference time
4. Epsilon prevents division by zero for near-constant inputs
5. Learnable gamma/beta allow the network to undo normalization if needed
6. Normalization stabilizes gradient flow through deep networks
7. Pre-Norm placement (before sub-layer) is standard in modern LLMs

Summary of Findings

LayerNorm Distribution

Normalizes each sample to mean~0, variance~1 along feature dim.

RMSNorm Distribution

Rescales to RMS~1 but does NOT subtract mean. Simpler, fewer ops.

LayerNorm vs RMSNorm

Highly correlated outputs. Main difference: mean centering.

Epsilon Effect

Too small eps can cause instability with near-constant inputs.

Learnable Parameters

gamma/beta can recover original distribution if the network learns to.

Gradient Flow

Normalization prevents vanishing/exploding gradients through deep layers.

Pre-Norm vs Post-Norm

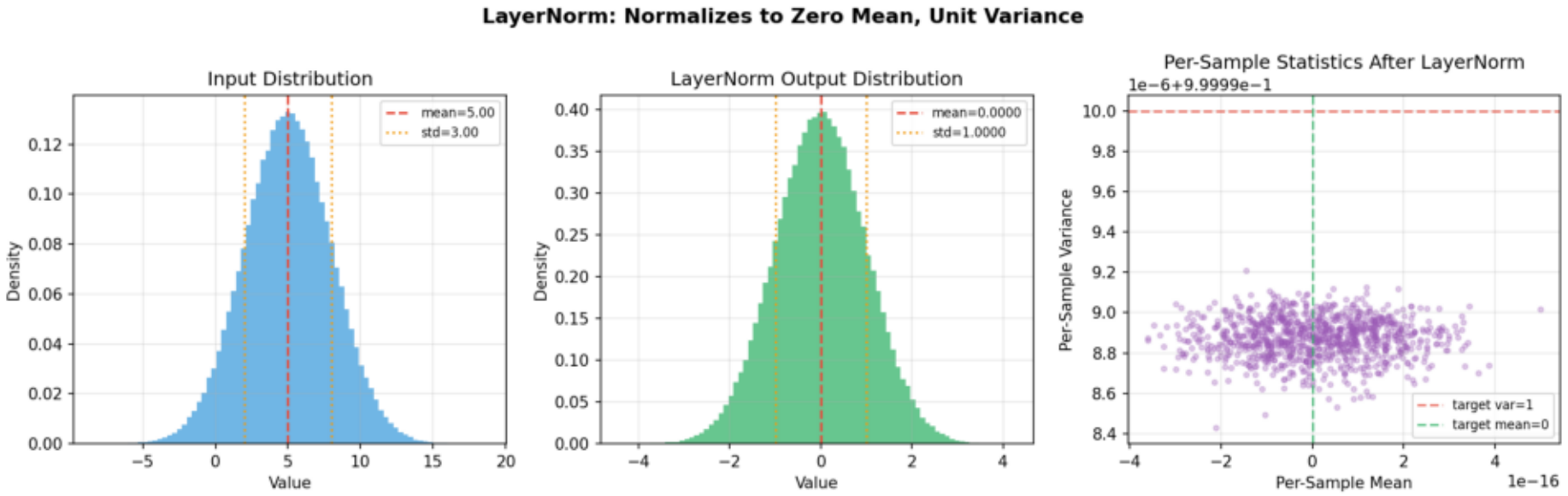
Pre-Norm creates a direct gradient highway; standard in modern LLMs.

3D Sequences

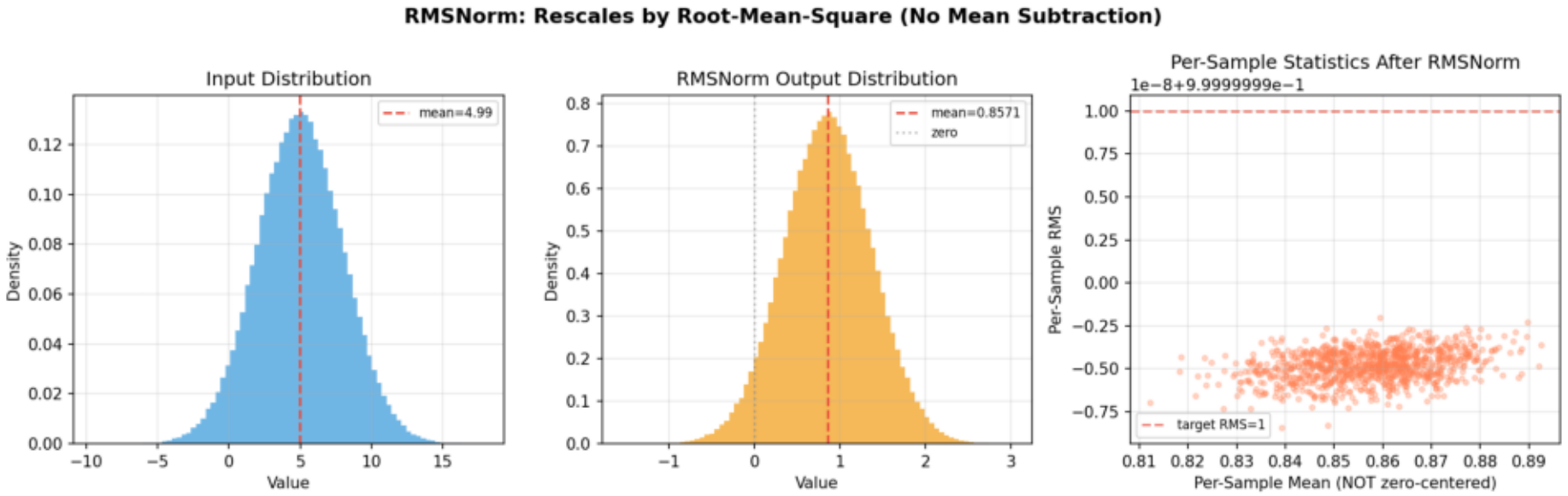
Both norms handle (B, L, D) naturally, normalizing per-position.

LayerNorm: GPT-2, GPT-3, BERT | *RMSNorm: LLaMA, Mistral, Gemma*

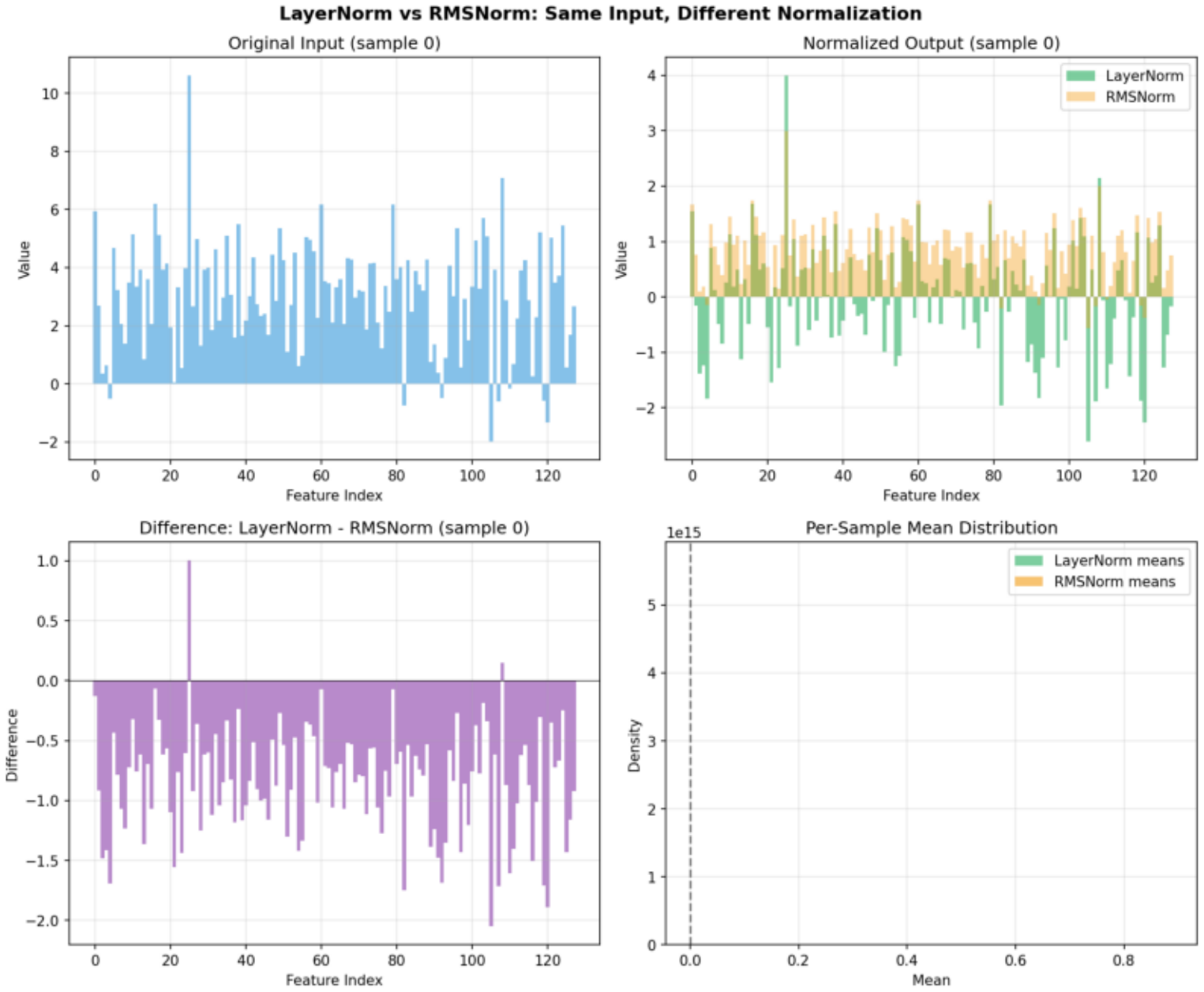
LayerNorm Input vs Output Distributions



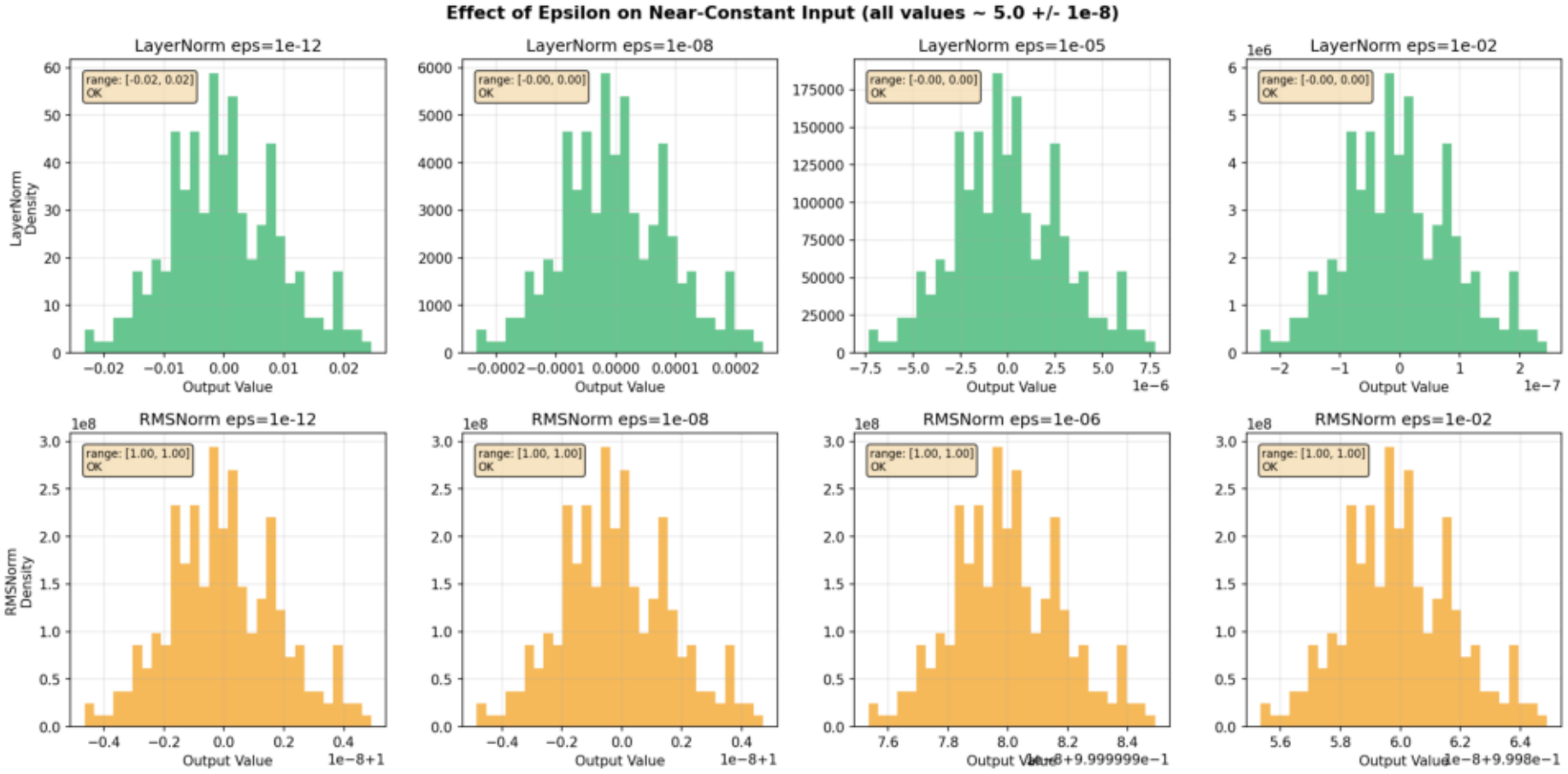
RMSNorm Input vs Output Distributions



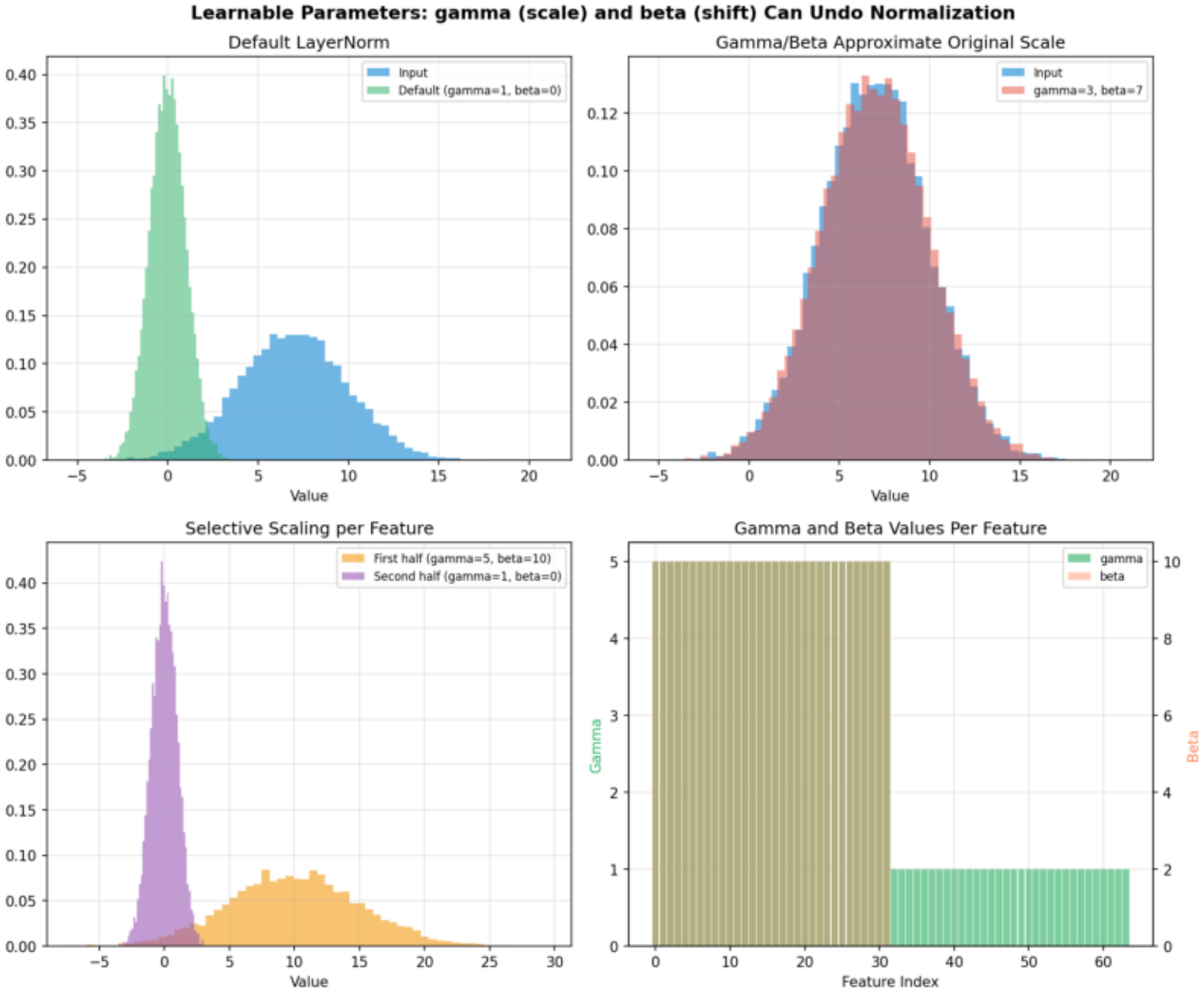
LayerNorm vs RMSNorm Comparison



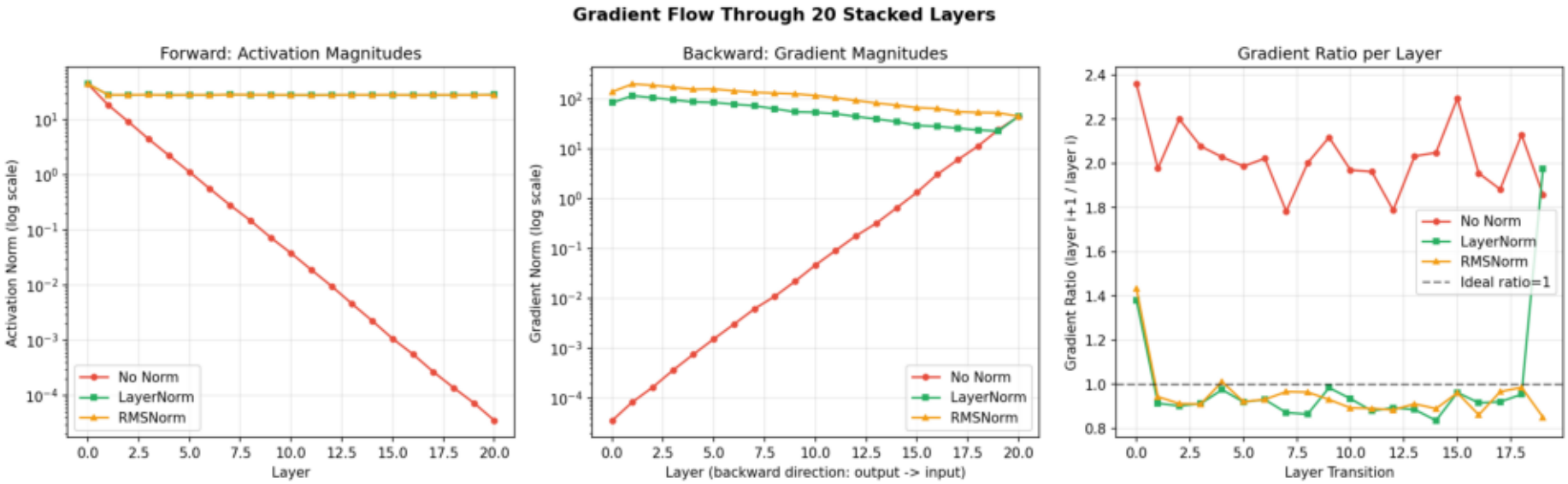
Effect of Epsilon on Near-Constant Input



Learnable Parameters (gamma/beta)



Gradient Flow: With vs Without Normalization



Pre-Norm vs Post-Norm Architecture

3D Sequence Normalization (Transformer-like)