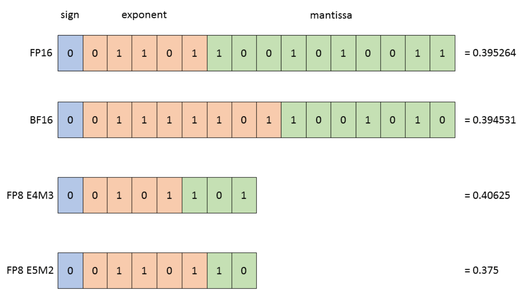
# Towards Fully FP8 LLM Training at Scale

A. Hernández-Cano[1*], D. Garbaya[1*], I. Schlag[2], M. Jaggi[1]

EPFL[1] · ETHZ[2]

## Context




FP8 format offers 2× theoretical speedup but suffers from:

- Limited dynamic range
- Large outlier activations
- Late-stage divergence issues

**Prior Work:** Fall back to BF16 attention or/and fine-grained kernels.

**Our Solution:** We introduce robust yet simple Transformer architectures that drastically reduce activation outliers enabling FP8 attention & MLP computations, without downstream performance tradeoff.

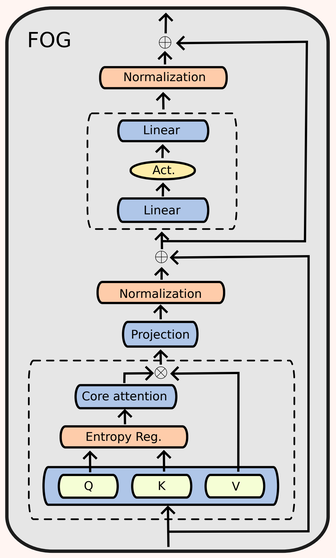
## FOG: Fast and Outlier-Guarded Architectures

A few simple modifications:

- Remove Pre-Norm
- Use Post-Norm
- Freeze QK regularization gains
- Upweight embeddings and use layerscale (good practices)

Three main variants:

- FOG-max: fast, highest downstream quality
- FOG-opt: fast, high quality
- FOG-flash: fastest, high quality

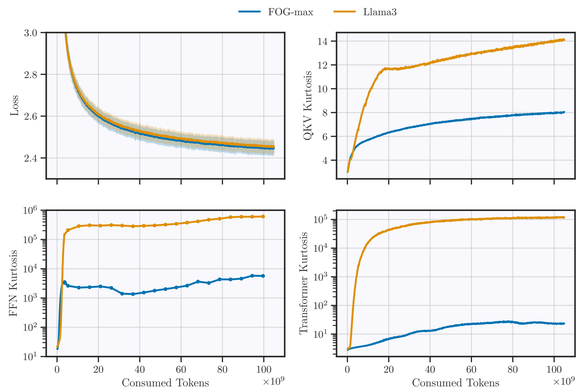



## Kurtosis and Stability

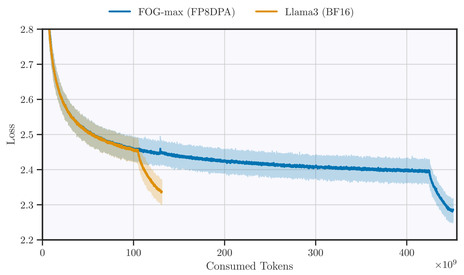
**Kurtosis as early warning metric:**

Kurtosis measures outlier presence: $\text{kurt}(x) = \frac{\mu[x^4]}{\sigma^2[x^2]}$

- **Tracked activations:** QKV projections, FFN inputs, block outputs
- **Signal:** Kurtosis can diverge much earlier than the loss
- **FOG advantage:** Orders of magnitude lower kurtosis vs. baseline
- **Stable patterns:** Sub-linear to logarithmic growth

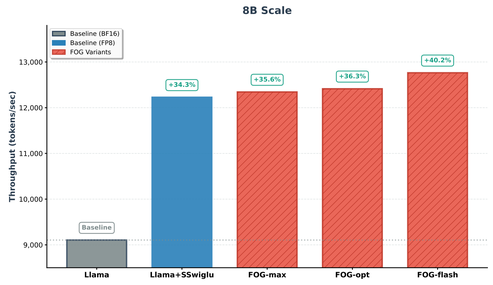



All baselines exhibit early divergence with FP8 attention. Conversely, we scaled a 1.5B FOG variant up to **450B tokens** (30x Chinchilla).

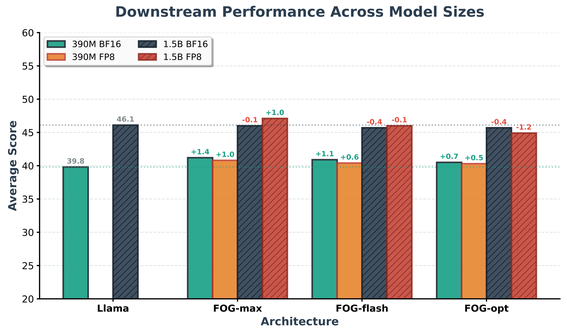



## Efficiency Gains

FOG variants achieve **unprecedented** throughput gains with high architectural flexibility, requiring only minimal tweaks.



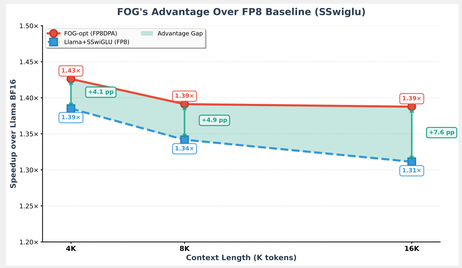

## Downstream Performance




FOG variants achieve parity with the BF16 baseline on downstream benchmarks while training significantly faster.

## Additional Results

**Long context**: the dot product attention tends to dominate FLOPs with long context. Computing it in FP8 yields a significant throughput boost.

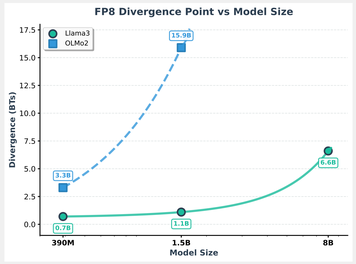



**MoEs**: We show FOG's stability, efficiency, and performance generalize to the MoE setting.

**Flexible choice of the activation function**: The proposed architectures support both pointwise activations (xIELU, GeLU) and gated activations (SwiGLU). Unlike baseline Llama that suffer from late-stage divergence because of SwiGLU's quadratic behaviour, FOG-max, using xIELU, remains stable throughout training.
We also demonstrate successful FOG-SwiGLU training.

**Memory efficiency**: FOG trains smoothly with FP8 optimizer states (instead of FP32), enabling considerable memory savings without compromising training stability.

**Intriguing pattern**:
Interestingly, larger models consistently diverge later than smaller models. What factors could account for this increased robustness?




NeurIPS 2025 - Advances in Neural Information Processing Systems