GIFT: Geometry-Informed Low-Precision Gradient Communication for LLM Pretraining

Jieying Wang
Department of Computer Science
Rutgers University
Piscataway, United States
jw2088@scarletmail.rutgers.edu

Shuyuan Fan
Department of Electrical and Computer Engineering
Rutgers University
Piscataway, United States
sf850@scarletmail.rutgers.edu

Mingkai Zheng
Department of Electrical and Computer Engineering
Rutgers University
Piscataway, United States
mz687@rutgers.edu

Fangru Linghu
Department of Electrical and Computer Engineering
Rutgers University
Piscataway, United States
fangru.linghu@rutgers.edu

Zhao Zhang
Department of Electrical and Computer Engineering
Rutgers University
Piscataway, United States
zz671@soe.rutgers.edu

***Abstract*—Gradient communication is a major scaling bottleneck in large language model (LLM) pretraining. While FP8 can substantially reduce communication cost, direct quantization in Euclidean coordinates can distort highly anisotropic gradients. We present GIFT, a geometry-informed gradient communication method that transforms gradients into more isotropic coordinates before low-precision quantization. To make this practical, GIFT uses a one-sided, low-rank geometry approximation and applies it selectively to layers most sensitive to quantization error, while leaving the optimizer, training recipe, communication collective, and FP8 format unchanged. Across Llama-style models with approximately 600M and 2B parameters, GIFT achieves downstream performance comparable to Muon training with FP32 gradients. On 64 NVIDIA GH200 Superchips, GIFT reduces end-to-end training step time by 10.87% for Llama-600M and 17.05% for Llama-2B relative to FP32 gradient communication.**

***Index Terms*—large language model pretraining, distributed training, gradient communication, FP8, information geometry, K-FAC**

I. Introduction

Distributed LLM pretraining synchronizes gradients on the critical path of every data-parallel iteration. FP8 communication reduces communicated gradient payload by 75% relative to FP32 [1], but this reduction is useful only if low-precision gradients preserve the training signal. Existing methods largely improve scaling or smooth numerical outliers while keeping the communication coordinates fixed. Our analysis instead identifies strong layer-wise anisotropy as an additional source of error: equal Euclidean perturbations can have very different effects on the model distribution, so quantization distortion depends on the coordinates in which gradients are represented. GIFT therefore treats the communication coordinate system itself as part of the quantization design.

II. Geometry-Informed Communication

We characterize local sensitivity with the Fisher metric. For a linear layer with weight gradient W_g, K-FAC [2] approximates the layer-wise Fisher block as

$$F_W \approx A \otimes G, \qquad A = \mathbb{E}[aa^\mathsf{T}], \quad G = \mathbb{E}[\delta\delta^\mathsf{T}]. \tag{1}$$

With $A = L_A L_A^\mathsf{T}$ and $G = L_G L_G^\mathsf{T}$, the corresponding dual gradient coordinates are

$$\widetilde{W}_g = L_G^{-1} W_g L_A^{-\mathsf{T}}. \tag{2}$$

FP8 communication is performed in these geometry-normalized coordinates, and the synchronized gradient is mapped back before the optimizer update. GIFT uses this geometry only for communication and does not modify the optimizer update.

III. Practical GIFT Design

The full two-sided transformation improves reconstruction fidelity but adds computation. GIFT therefore retains only the input-side geometry and further approximates it with a rank-32 low-rank-plus-diagonal factor,

$$\widetilde{A}_r = U_r \Lambda_r U_r^\mathsf{T} + D_r = L_{A,r} L_{A,r}^\mathsf{T}, \qquad r = 32. \tag{3}$$

In controlled FP8 round-trip experiments, Euclidean FP8 has RelL2 5.272×10^{-2}, dense input-side geometry reduces it to 2.223×10^{-2}, and the rank-32 approximation achieves 2.494×10^{-2}. Thus, the practical rank-32 design reduces RelL2 by 52.7% relative to Euclidean FP8 and retains approximately 91% of the error-reduction benefit of dense input-side geometry.

GIFT is applied only where geometry-aware communication is most beneficial. Before formal pretraining, a separate 200-step Layer-wise Euclidean FP8 profiling run records cumulative FP8 boundary hits and quantization error for candidate MLP layers. Layers are ranked by boundary hits, and GIFT selects the smallest prefix whose cumulative quantization error accounts for at least 30% of the total. The selected set is then fixed for a fresh pretraining run.

For selected layers, geometry statistics are refreshed every $K = 50$ steps by synchronizing input-side covariance statistics across the data-parallel group, after which each rank locally reconstructs the same rank-32 factor. Error feedback is maintained in the transformed coordinates and reset when the

geometry is refreshed. GIFT uses E5M2 FP8 with a MAX-synchronized scale across data-parallel ranks.

IV. Evaluation

We implement GIFT in Megatron-LM [3] and pretrain Llama-600M and Llama-2B on OpenWebText for 60K steps using Muon. GIFT improves downstream-task preservation over direct Layer-wise Euclidean FP8: it matches or exceeds the Muon FP32 baseline on 8 of 14 tasks for Llama-600M and 7 of 14 tasks for Llama-2B, compared with 5 of 14 and 4 of 14 tasks, respectively, for Euclidean FP8.

On 64 NVIDIA GH200 Superchips, GIFT reduces end-to-end training step time by 10.87% for Llama-600M and 17.05% for Llama-2B relative to FP32 gradient communication. Despite the additional geometry processing, GIFT retains essentially the full 75% gradient-payload reduction of FP8. These results show that adapting communication coordinates to local gradient geometry improves the fidelity–efficiency trade-off of low-precision distributed training.

V. Conclusion

GIFT reframes low-precision gradient communication as both a numerical-format problem and a coordinate-system problem. A selective input-side, rank-32 geometry transformation substantially improves FP8 gradient fidelity and downstream-task preservation while retaining meaningful end-to-end speedups at scale.

References

[1] P. Micikevicius et al., "FP8 Formats for Deep Learning," *arXiv preprint arXiv:2209.05433*, 2022.

[2] J. Martens and R. Grosse, "Optimizing Neural Networks with Kronecker-Factored Approximate Curvature," in *Proc. 32nd Int. Conf. Machine Learning (ICML)*, 2015, pp. 2408–2417.

[3] D. Narayanan et al., "Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM," in *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC)*, 2021.