Low-Rank Curvature for Zeroth-Order Optimization in LLM Fine-Tuning

Hyunseok Seung[1] **Jaewoo Lee**[2] **Hyunsuk Ko**[3]

[1]University of Wisconsin – Madison [2]University of Georgia [3]Hanyang University

Paper Code

Motivation and Overview

- **Memory Bottleneck in LLM Fine-Tuning**
  Fine-tuning LLMs via backpropagation incurs substantial memory overhead, motivating zeroth-order (ZO) methods that rely solely on forward passes.
- **Limitations of ZO Methods**
  Finite-difference ZO gradients have high variance, and isotropic perturbations ignore anisotropic curvature, causing slow and unstable optimization.
- **A Curvature-Aware, Variance-Reduced ZO Method**
  We propose **LOREN**, a ZO optimizer that learns a low-rank perturbation covariance and integrates REINFORCE Leave-One-Out (RLOO) variance reduction into gradient estimation.

Curvature-Aware Zeroth-Order Gradients

- **ZO Gradient Estimation**

$$\hat{\nabla} f(\mathrm{x}) = \mathbb{E}_{\mathrm{u}\sim\mathcal{N}(0,\,\mathrm{I})}\left[\frac{f(\mathrm{x}+\epsilon\mathrm{u}) - f(\mathrm{x})}{\epsilon}\,\mathrm{u}\right] \in \mathbb{R}^{m\times n}.$$

- **Preconditioning ⇔ Covariance Learning**

$$\mathrm{H}^{-1}\nabla f(\mathrm{x}) \approx \mathbb{E}_{\tilde{\mathrm{u}}\sim\mathcal{N}(0,\,\mathrm{H}^{-1})}\left[\frac{f(\mathrm{x}+\epsilon\tilde{\mathrm{u}}) - f(\mathrm{x})}{\epsilon}\,\tilde{\mathrm{u}}\right].$$

- **Memory-Efficient Low-Rank Covariance Modeling**

$$\tilde{\mathrm{H}} = \mathrm{I}_m \otimes (\rho \mathrm{I}_n + \mathrm{a}\mathrm{a}^\top), \qquad \mathrm{a} \in \mathbb{R}^n.$$

LOREN Algorithm

Initialize
model params x_0
$\mathrm{a}_0 \sim \mathcal{N}(0,\ \mathrm{I})$

Forward Passes
for $k = 1, \dots, K$,
$\mathrm{u}_k \sim \mathcal{N}(0, \mathrm{I})$
$f^k \leftarrow f(\mathrm{x}_t + \epsilon\,\Sigma^{1/2}\mathrm{u}_k)$

Variance-Reduced Preconditioned ZO Gradients

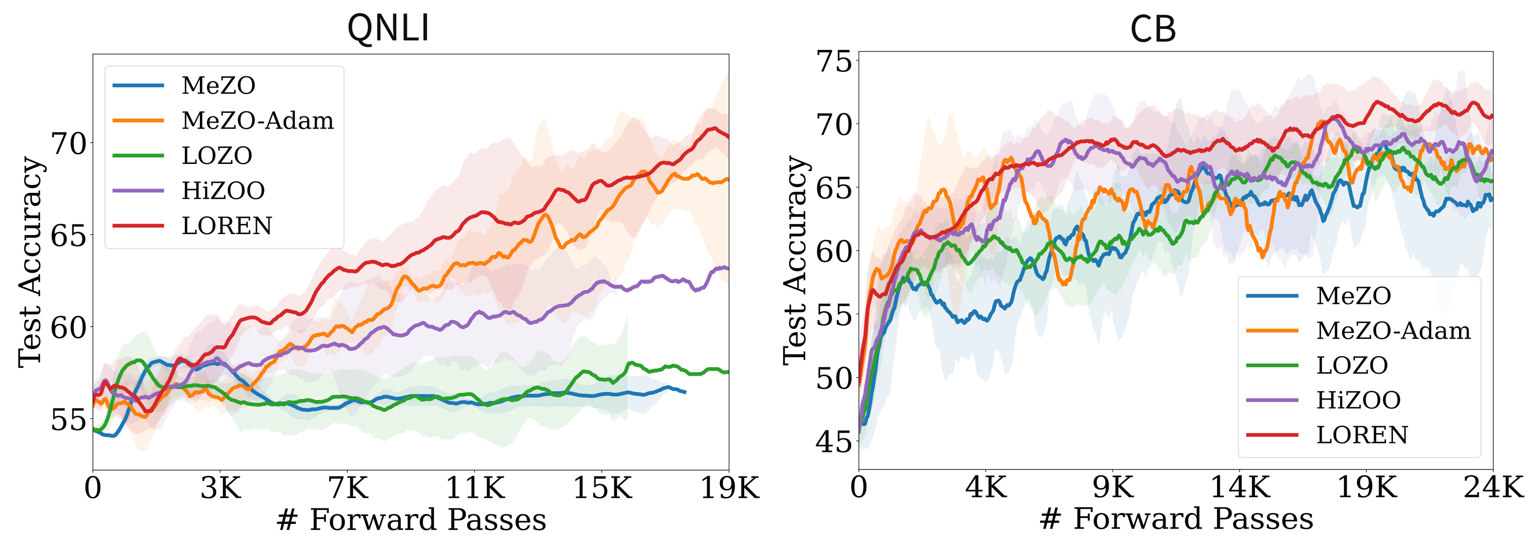
$$\mathrm{g}(\mathrm{x}_t) \leftarrow \frac{1}{\epsilon(K-1)}\sum_{k=1}^{K}\left(f^k - \frac{1}{K}\sum_{j=1}^{K} f^j\right)\Sigma^{1/2}\mathrm{u}_k$$

$$\mathrm{g}(\mathrm{a}_t) \leftarrow \frac{1}{K-1}\sum_{k=1}^{K}\left(f^k - \frac{1}{K}\sum_{j=1}^{K} f^j\right)\nabla_{\mathrm{a}}\log\mathcal{N}(\mathrm{x}, \epsilon^2\Sigma)$$

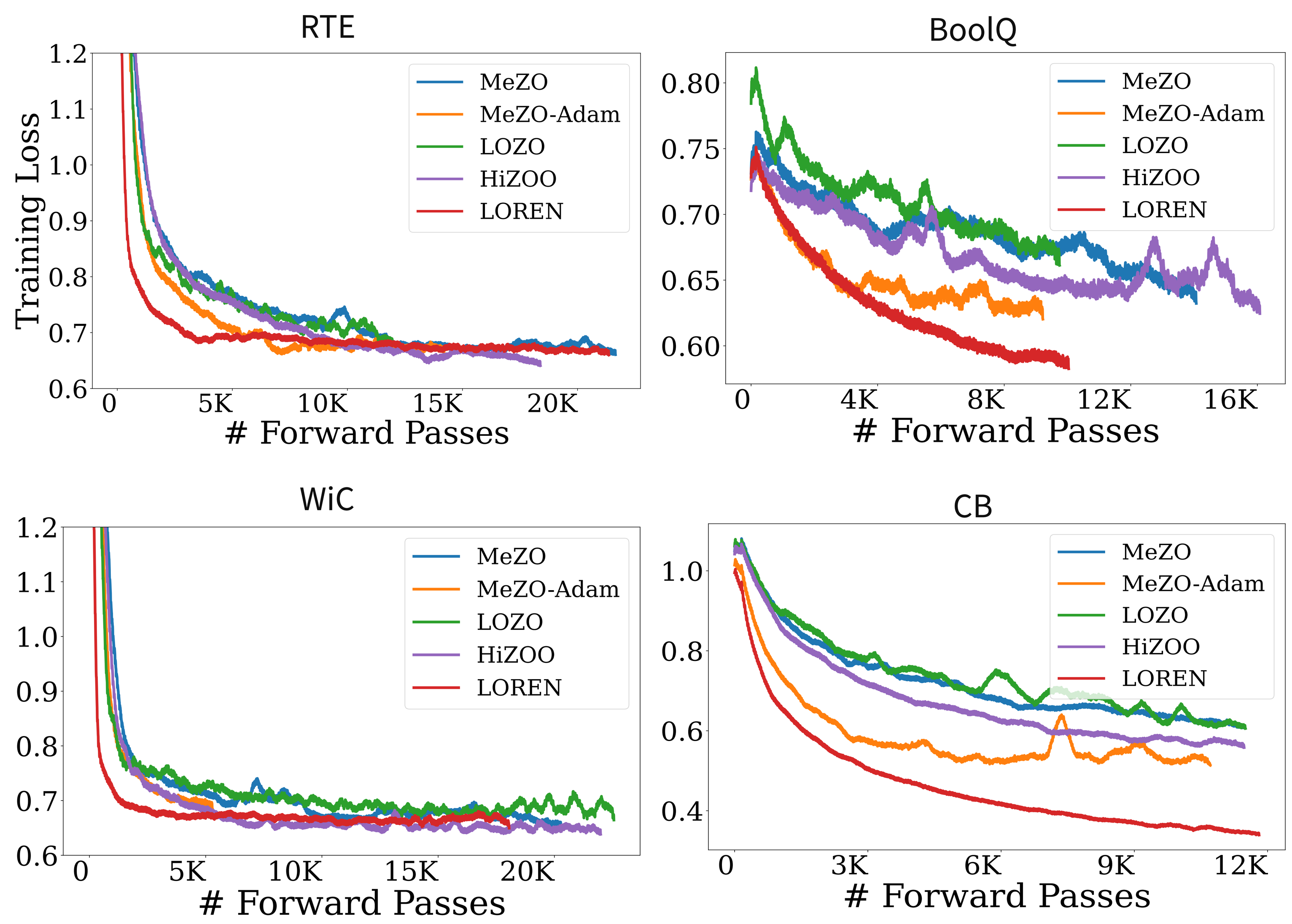
Update
$\mathrm{x}_{t+1} \leftarrow \mathrm{x}_t - \eta\,\mathrm{g}(\mathrm{x}_t)$
$\mathrm{a}_{t+1} \leftarrow \mathrm{a}_t - \nu\,\mathrm{g}(\mathrm{a}_t)$

Results on LLM Fine-Tuning

- **Test Accuracy (GPT-2, OPT-13B)**



- **Training Loss (OPT-13B)**



✓ LOREN consistently achieves **faster convergence** and **higher accuracy** than other ZO methods under the same query budget.

- **Memory Footprint**

	GPT-2-XL	LLaMA-3-8B	OPT-13B
MeZO	16.9 (1.00×)	18.4 (1.00×)	32.9 (1.00×)
MeZO-Adam	28.8 (1.70×)	46.2 (2.51×)	76.0 (2.31×)
MeZO-SVRG	32.3 (1.91×)	44.3 (2.41×)	74.7 (2.27×)
LOZO	16.8 (0.99×)	17.4 (0.95×)	32.8 (1.00×)
HiZOO	24.3 (1.44×)	36.5 (1.98×)	59.6 (1.81×)
LOREN	23.1 (1.37×)	33.6 (1.83×)	57.5 (1.75×)

✓ LOREN still maintains the **affordable memory usage.**

- **Training Efficiency**

	GPT-2-XL		LLaMA-3-8B	
	# Queries	Time (hr)	# Queries	Time (hr)
MeZO	16,752	3.13	5,736	0.80
MeZO-Adam	1,632	**0.32**	6,894	0.96
MeZO-SVRG	4,248	1.51	3,216	0.53
LOZO	10,686	0.66	3,198	**0.15**
HiZOO	2,232	0.98	4,356	0.76
LOREN	**1,320**	0.33	**1,512**	0.47

✓ LOREN reaches target accuracy with the **fewest queries** while remaining competitive in wall-clock time.