NEURAL INFORMATION
PROCESSING SYSTEMS

# Zeroth-Order Optimization Finds Flat Minima


**Liang Zhang**

(ETH Zurich & Max Planck Institute)

**Bingcong Li**

(ETH Zurich)

**Kiran Thekumparampil**

(Amazon)

**Sewoong Oh**

(University of Washington)

**Michael Muehlebach**

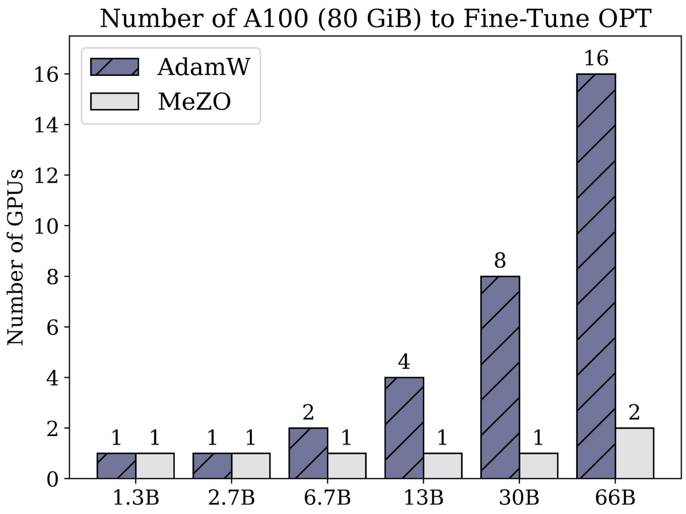
(Max Planck Institute)

**Niao He**

(ETH Zurich)

# Zeroth-Order Optimization is Extensively Applied

- Gradients are expensive to compute




**Fine-Tuning Large Language Models**

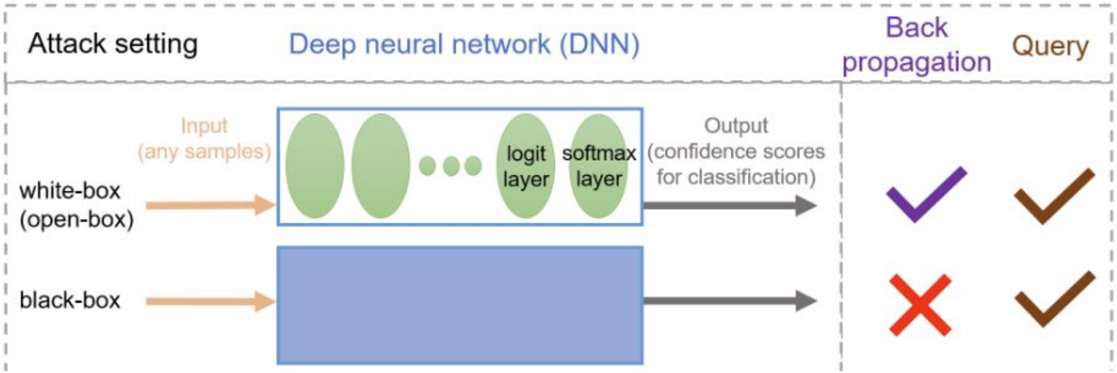
✘ Backpropagation heavy in **memory**

✔ MeZO [1]: zeroth-order methods with **only forward passes**

[1] Malladi et al. Fine-Tuning Language Models with Just Forward Passes. *NeurIPS*, 2023.

# Zeroth-Order Optimization is Extensively Applied

- Gradients are expensive to compute (**Fine-Tuning Large Language Models**)
- Gradients are infeasible




**Black-Box Attacks** [2]

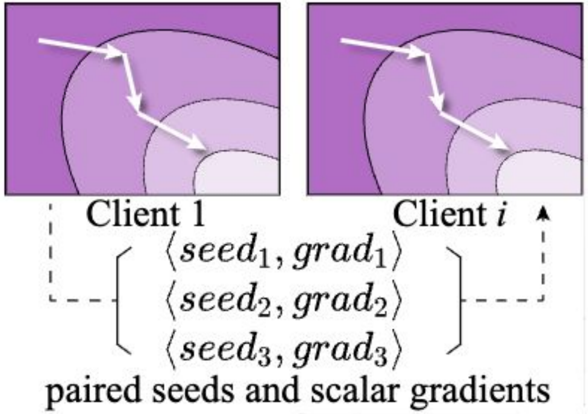

[2] Chen et al. ZOO: Zeroth Order Optimization Based Black-Box Attacks to Deep Neural Networks without Training Substitute Models. *ACM*, 2017.

# Zeroth-Order Optimization is Extensively Applied

- Gradients are expensive to compute (**Fine-Tuning Large Language Models**)
- Gradients are infeasible (**Black-Box Attacks**)
- Save communication costs

**Federated Learning** [3]

Communication cost: O(1) v.s. O($d$)




[3] Qin et al. Federated Full-Parameter Tuning of Billion-Sized Language Models with Communication Cost under 18 Kilobytes. *ICML*, 2024.

# Zeroth-Order Optimization is Extensively Applied

- Gradients are expensive to compute (**Fine-Tuning Large Language Models**)
- Gradients are infeasible (**Black-Box Attacks**)
- Save communication costs (**Federated Learning**)
- and more …

# Zeroth-Order Optimization: What We Know

For the problem $\min_{x \in \mathbb{R}^d} f(x)$

$$g_\lambda(x_t, u_t) = \frac{f(x_t + \lambda u_t) - f(x_t - \lambda u_t)}{2\lambda} u_t.$$

$$x_{t+1} \leftarrow x_t - \eta \, g_\lambda(x_t, u_t).$$

converges with $\mathbb{E}[f(\bar{x}_T) - \min_{x \in \mathbb{R}^d} f(x)] \leq \mathcal{O}(d/T)$ when convex and smooth [4]

[4] Nesterov and Spokoiny. Random Gradient-Free Minimization of Convex Functions. *Foundations of Computational Mathematics*, 2017.

# Zeroth-Order Optimization: What We Know

For the problem $\min_{x \in \mathbb{R}^d} f(x)$

$$g_\lambda(x_t, u_t) = \frac{f(x_t + \lambda u_t) - f(x_t - \lambda u_t)}{2\lambda} u_t.$$

$$x_{t+1} \leftarrow x_t - \eta\, g_\lambda(x_t, u_t).$$

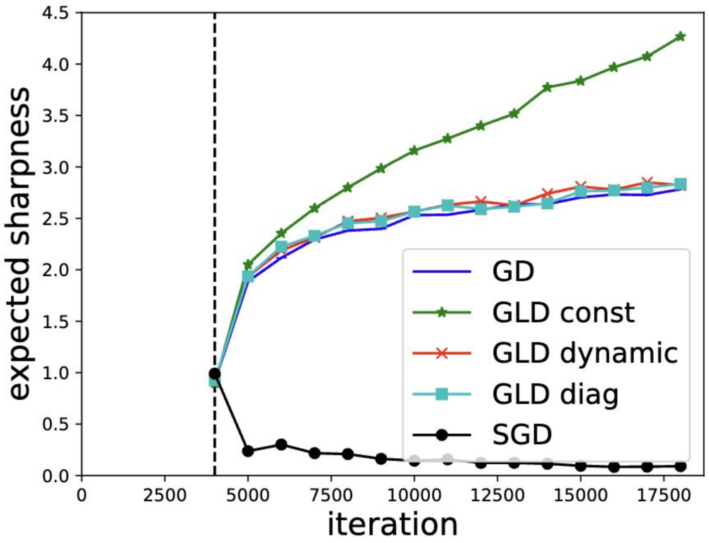
converges with $\mathbb{E}[f(\bar{x}_T) - \min_{x \in \mathbb{R}^d} f(x)] \leq \mathcal{O}(d/T)$ when convex and smooth [4]

**but which minima in the set of minimizers?**

[4] Nesterov and Spokoiny. Random Gradient-Free Minimization of Convex Functions. *Foundations of Computational Mathematics*, 2017.

# Existing Study on Implicit Regularization: Mostly First-Order

- SGD converges to solutions with small expected sharpness (trace of Hessian)




**VGG11 on CIFAR-10** [5]

Expected sharpness:

$$\mathbb{E}_{u \sim \mathcal{N}(0, \mathrm{I}_d)}[f(x+\delta u)] - f(x)$$

Average over 100 samples and with

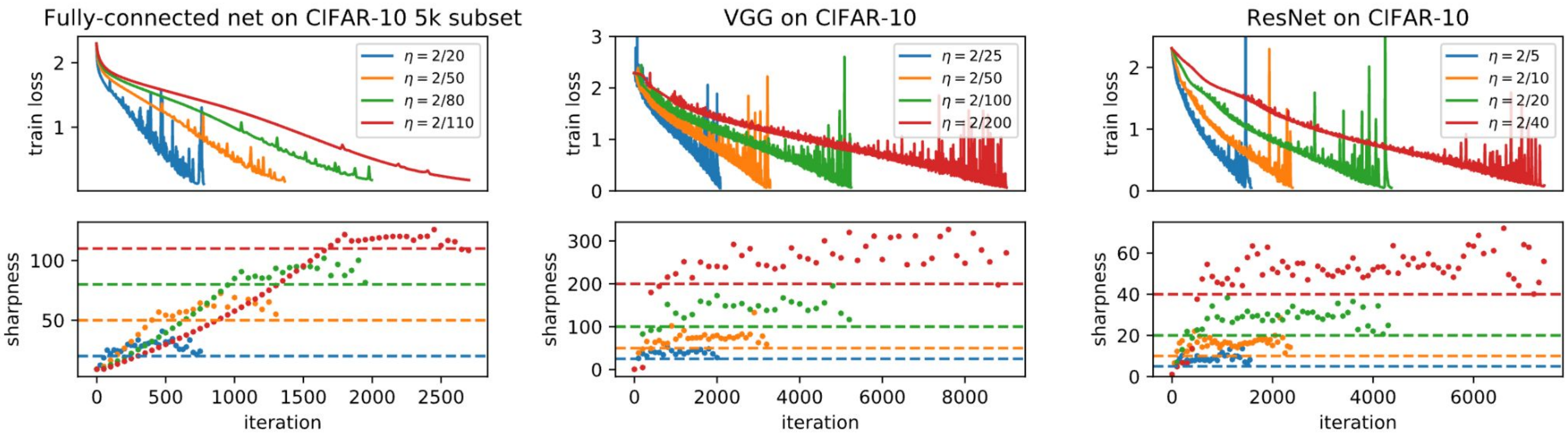
$$\delta = 0.01$$

[5] Zhu et al. The Anisotropic Noise in SGD: Its Behavior of Escaping from Sharp Minima and Regularization Effects. *ICML*, 2019.

# Existing Study on Implicit Regularization: Mostly First-Order

- SGD converges to solutions with small expected sharpness (trace of Hessian)
- (S)GD with large stepsizes penalizes largest eigenvalue of Hessian

**Edge of Stability** [6]





[6] Cohen et al. Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability. *ICLR*, 2021.

# Existing Study on Implicit Regularization: Mostly First-Order

- SGD converges to solutions with small expected sharpness (trace of Hessian)
- (S)GD with large stepsizes penalizes largest eigenvalue of Hessian
- SGD with label noise decreases trace of Hessian [7]
- SAM [8] minimizes trace of Hessian [9,10] or largest eigenvalue of Hessian [10, 11]

[7] Li et al. What Happens after SGD Reaches Zero Loss? – A Mathematical Framework. *ICLR*, 2022.
[8] Foret et al. Sharpness-Aware Minimization for Efficiently Improving Generalization. *ICLR*, 2021.
[9] Ahn et al. How to Escape Sharp Minima with Random Perturbations. *ICML*, 2024.
[10] Wen et al. How Sharpness-Aware Minimization Minimizes Sharpness? *ICLR*, 2023.
[11] Bartlett et al. The Dynamics of Sharpness-Aware Minimization: Bouncing Across Ravines and Drifting Towards Wide Minima. *JMLR*, 2023.

# Zeroth-Order Optimization Minimizes Trace of Hessian

Unbiased gradient estimator for the **smoothed function** $f_\lambda(x) := \mathbb{E}_{u \sim \mathcal{N}(0, \mathrm{I}_d)}[f(x + \lambda u)]$

$$\mathbb{E}[g_\lambda(x, u)] = \nabla f_\lambda(x)$$

Taylor's theorem gives

$$f(x + \lambda u) = f(x) + \lambda u^\top \nabla f(x) + \frac{\lambda^2}{2} u^\top \nabla^2 f(x) u + o(\lambda^2)$$

Taking expectation,

$$\begin{aligned} f_\lambda(x) &= f(x) + \frac{\lambda^2}{2} \mathbb{E}_u \left[ \mathrm{Tr} \left( u u^\top \nabla^2 f(x) \right) \right] + o(\lambda^2) \\ &= f(x) + \frac{\lambda^2}{2} \mathrm{Tr} \left( \nabla^2 f(x) \right) + o(\lambda^2). \end{aligned}$$

# Zeroth-Order Optimization Minimizes Trace of Hessian

Unbiased gradient estimator for the **smoothed function** $f_\lambda(x) := \mathbb{E}_{u\sim\mathcal{N}(0,\mathrm{I}_d)}[f(x+\lambda u)]$

$$\mathbb{E}[g_\lambda(x,u)] = \nabla f_\lambda(x)$$

Taylor's theorem gives

$$f(x+\lambda u) = f(x) + \lambda u^\top \nabla f(x) + \frac{\lambda^2}{2} u^\top \nabla^2 f(x) u + o(\lambda^2)$$

Taking expectation,

$$\begin{aligned} f_\lambda(x) &= f(x) + \frac{\lambda^2}{2}\mathbb{E}_u\left[\mathrm{Tr}\left(uu^\top \nabla^2 f(x)\right)\right] + o(\lambda^2) \\ &= f(x) + \frac{\lambda^2}{2}\boxed{\mathrm{Tr}\left(\nabla^2 f(x)\right)} + o(\lambda^2). \end{aligned}$$

**Trace of Hessian** as additional regularization!

# Definition of Flat Minima

- Set of minimizers $\mathcal{X}^* := \arg\min_{x \in \mathbb{R}^d} f(x)$
- Flat minima $x^* \in \arg\min_{x \in \mathcal{X}^*} \mathrm{Tr}(\nabla^2 f(x))$

$$\min_{x \in \mathbb{R}^d} \mathrm{Tr}(\nabla^2 f(x)), \quad \text{s.t.} \quad f(x) - \min_{x \in \mathbb{R}^d} f(x) \leq 0$$

- Approximate flat minima

$$f(\hat{x}) - \min_{x \in \mathbb{R}^d} f(x) \leq \epsilon_1, \quad \mathrm{Tr}(\nabla^2 f(\hat{x})) - \min_{x \in \mathcal{X}^*} \mathrm{Tr}(\nabla^2 f(x)) \leq \epsilon_2$$

# Complexity for Finding Flat Minima

Choosing number of iterations and stepsize as

$$T = \mathcal{O}(d^4/\epsilon^2) \qquad\qquad \lambda = \mathcal{O}(\epsilon^{1/2}/d^{3/2})$$

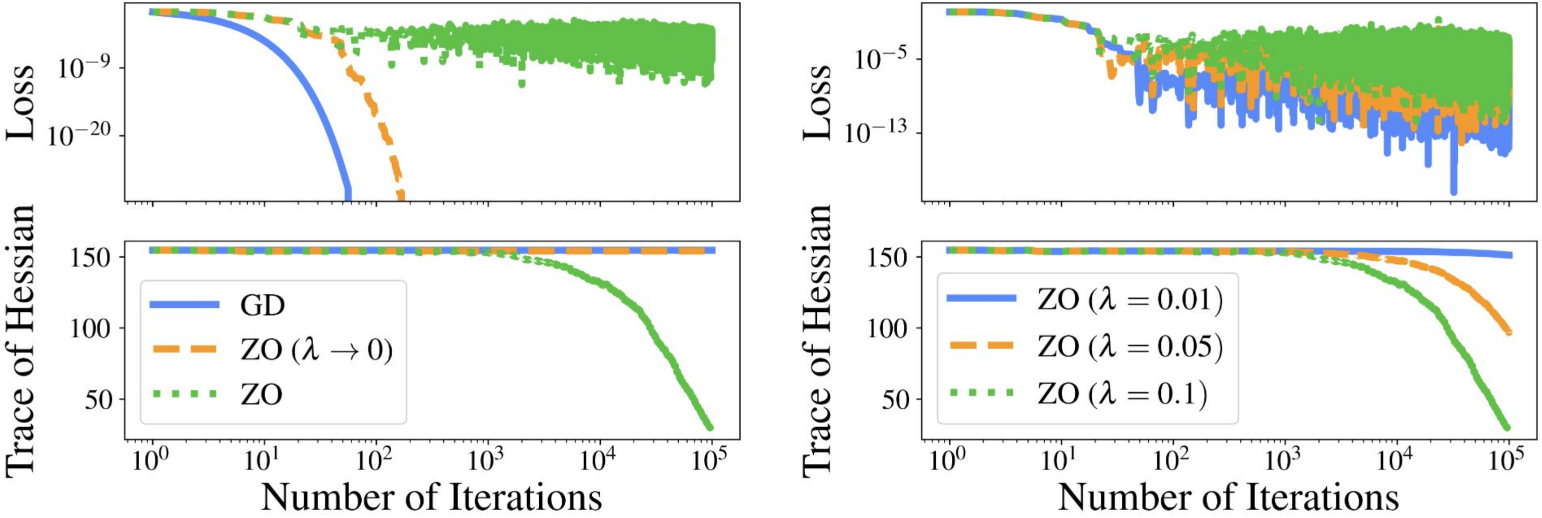
we have the guarantee

$$\mathbb{E}\left[f(x_\tau) - \min_{x\in\mathbb{R}^d} f(x)\right] \leq \ \mathcal{O}(\epsilon/d^2)$$

$$\mathbb{E}\left[\mathrm{Tr}(\nabla^2 f(x_\tau)) - \min_{x\in\mathcal{X}^*} \mathrm{Tr}(\nabla^2 f(x))\right] \leq \epsilon$$

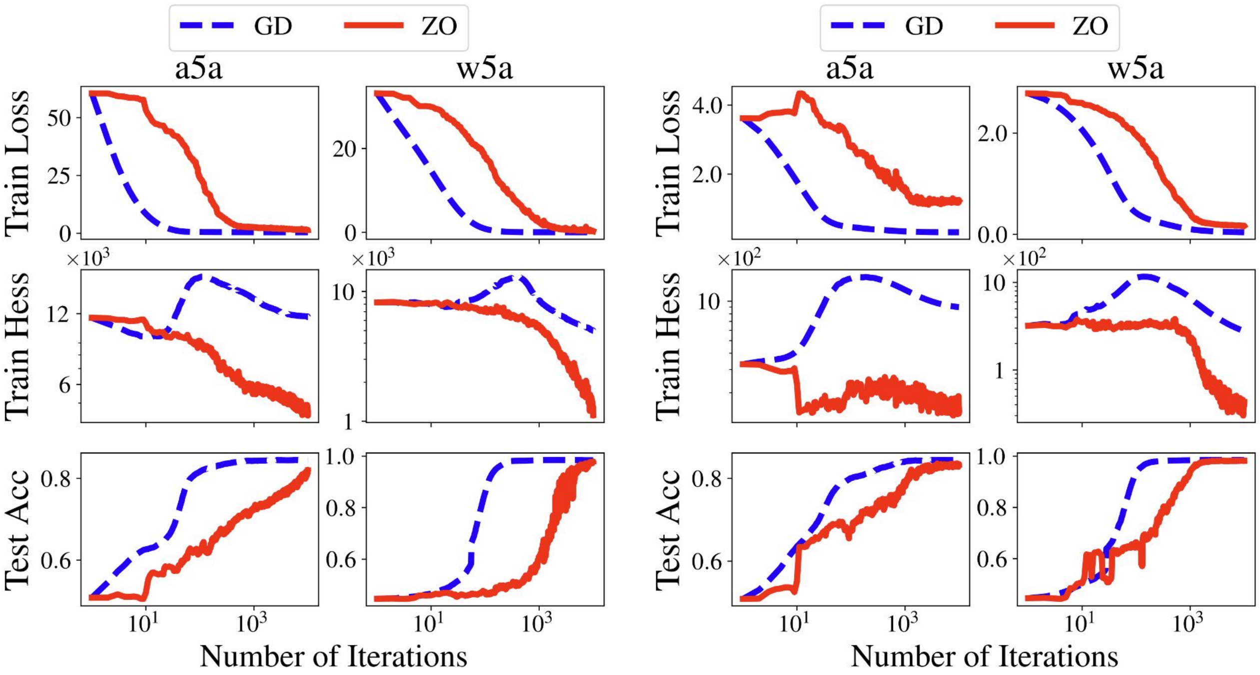
# Experiments on Test Function

Consider the function $(y^\top z - 1)^2/2$



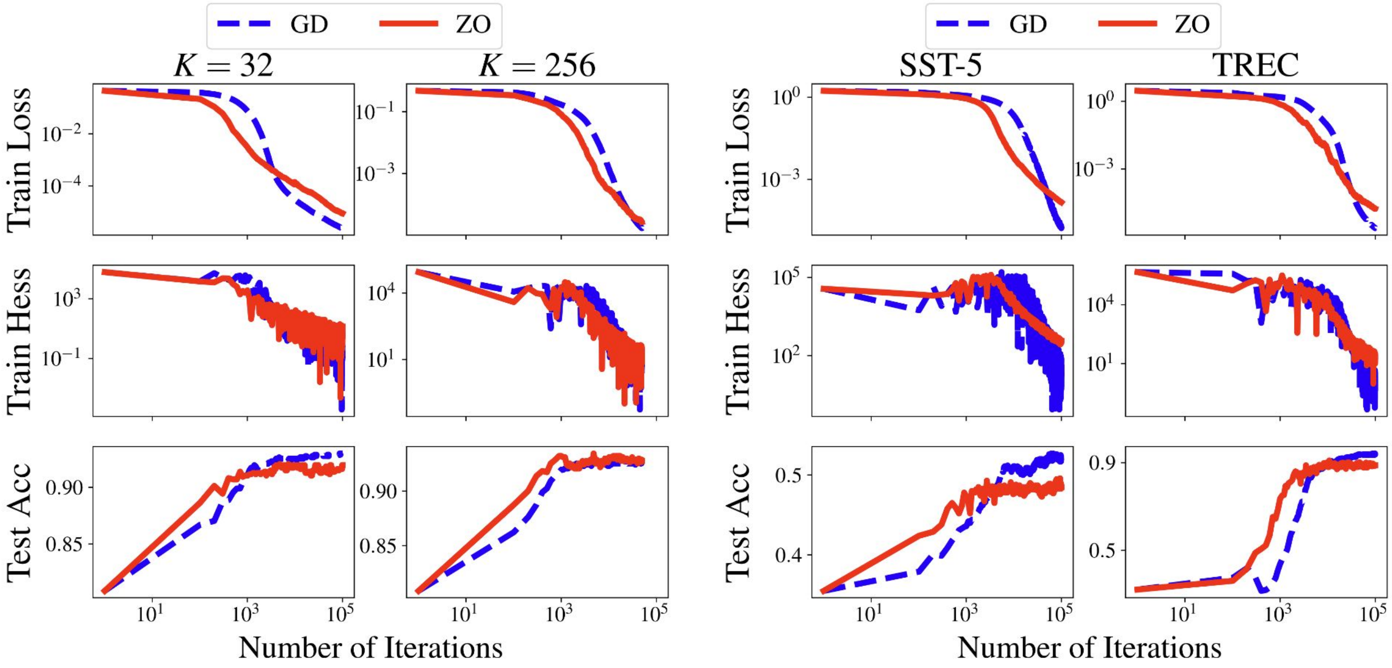

(a) Comparison of GD and ZO. (b) Effect of $\lambda$.

# Experiments on Binary Classification Tasks




(a) SVMs. (b) Logistic Regression.

# Experiments on Fine-Tuning Language Models




(a) SST-2.

(b) SST-5 and TREC.