DeepZero: Scaling up Zeroth-Order Optimization for Deep Model Training

Aochuan Chen[1,*] Yimeng Zhang [1,*]
Jinghan Jia[1], James Diffenderfer[2], Jiancheng Liu[2], Konstantinos Parasyris[2], Yihua Zhang[2], Zheng Zhang[3], Bhavya Kailkhura[2], Sijia Liu[1]

MICHIGAN STATE UNIVERSITY

[1] Michigan State University,
[2] Lawrence Livermore National Laboratory,
[3] UC Santa Barbara

Lawrence Livermore National Laboratory

OPTML
WE MAKE THE GLOBAL OPTIMAL

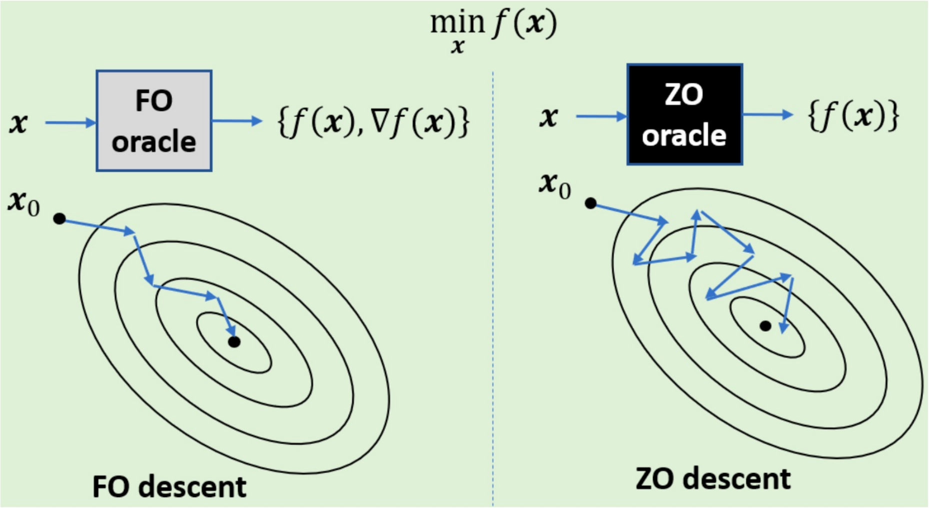
*Equal contributions

ICLR

MICHIGAN STATE UNIVERSITY

What is ZO Optimization?

ZO Optimization:
Gradient-free optimization that leverages **finite differences of function values to estimate gradients**, rather than requesting explicit gradient information



Advantages:

- Simple, easy to implement
- Provable convergence as first-order optimization

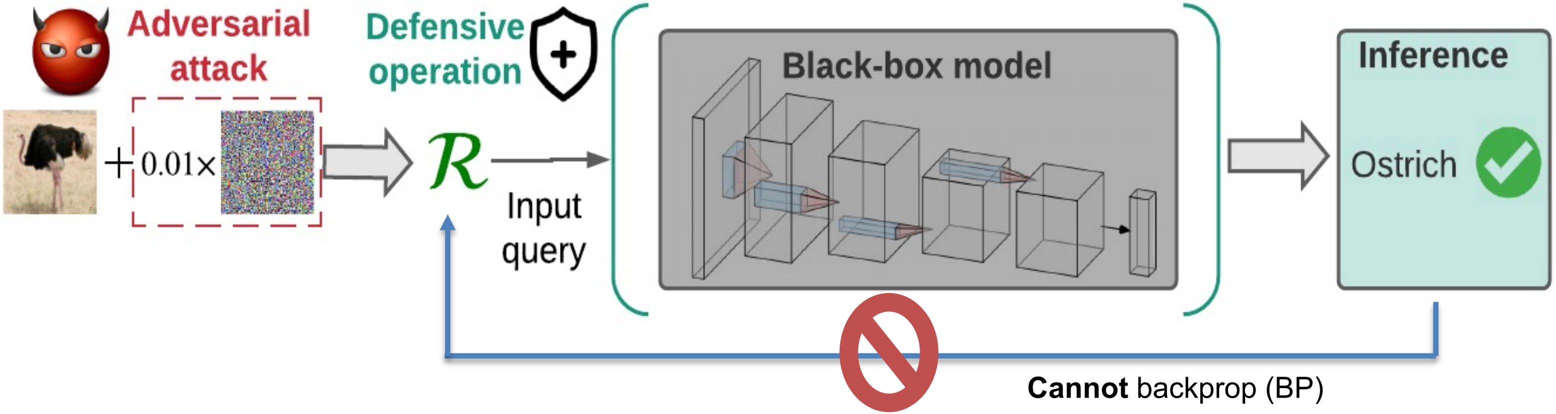
Challenges:

- Slow convergence
- Lack of scalability in high dimensions

Liu, et al. "A primer on zeroth-order optimization in signal processing and machine learning", IEEE Signal Processing Magazine, 2020

Why ZO Optimization?
"Robustifying" Black-Box ML Models

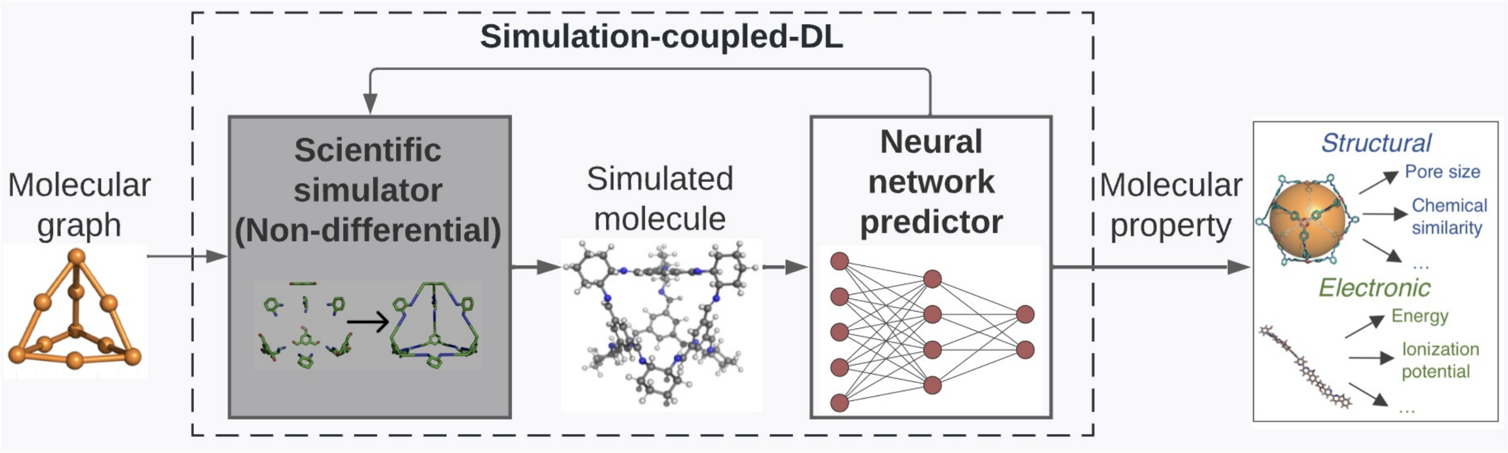
- **Robustifying "black-box" DL models** against adversarial attacks:



Zhang, Liu, et al. "How to robustify black-box ml models?" ICLR'22

Why ZO Optimization?
Simulation-Coupled DL in AI for Science

- **Simulation-coupled DL:** DL model integrated with non-differential simulators



Ioannis, et al. "Zeroth-Order SciML: Non-intrusive Integration of Scientific Software with Deep Learning." arXiv preprint arXiv:2206.02785 (2022).v

Challenge: Stateful ZO Methods Are Still Not Easy to Scale to DL Training from "Scratch"

Review of Stateful ZO Methods

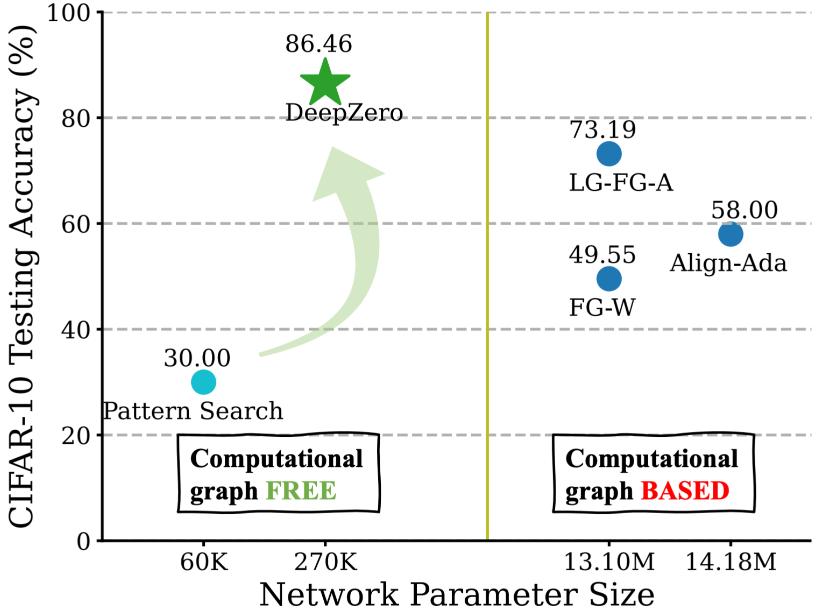
- **Pure ZO optimization:**

  - Using only model queries

- **BP-free but computation graph-based:**

  - forward gradients-based methods, LG-FG-A and FG-W (Ren et al., 2023),

  - input-weight alignment , Align-ada (Boopathy & Fiete, 2022)



M. Ren, S. Kornblith, R. Liao, and G. Hinton. "Scaling forward gradient with local losses." *ICLR'23*
A. Boopathy and I. Fiete. How to train your wide neural network without backprop: An input-weight alignment perspective. *ICML'22*

ZO Gradient Estimator: RGE or CGE?

Randomized Gradient Estimate (RGE)

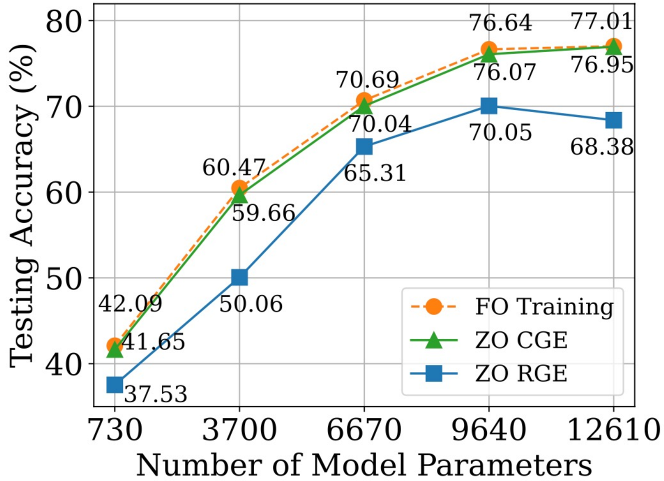
$$\hat{\nabla}_{\mathbf{w}} \ell(\mathbf{w}) = \frac{1}{q} \sum_{i=1}^{q} \left[\frac{d}{\mu} \left(\ell(\mathbf{w} + \mu \mathbf{u}_i) - \ell(\mathbf{w}) \right) \mathbf{u}_i \right]$$

Coordinate-wise Gradient Estimate (CGE)

$$\hat{\nabla}_{\mathbf{w}} \ell(\mathbf{w}) = \sum_{i=1}^{d} \left[\frac{\ell(\mathbf{w} + \mu \mathbf{e}_i) - \ell(\mathbf{w})}{\mu} \mathbf{e}_i \right],$$

$\ell(w)$	:	black-box function
w	:	the ***d*-dimension** parameter
$\{\mathbf{u}_i\}_{i=1}^{q}$	:	q random vectors
μ	:	step size, known as smoothing parameter
$e_i \in R^d$	:	ith elementary basis vector (1 at the ith coordinate and 0s elsewhere)

	CGE	RGE
Query efficiency (q < d)		😃
Computation efficiency	😃	
Accuracy (even q = d)	😃	

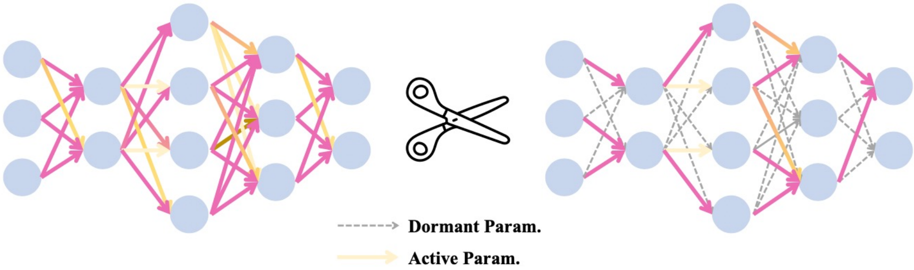


(CNN, CIFAR-10)

Pruning via ZO Oracle

- **Reducing query complexity of CGE** via "pruned gradients"
- **Proposed technique**: Model pruning via ZO oracle

Model Pruning



Sparse mask via **ZO gradient signal preservation (ZO-GraSP)**

$$\hat{\mathbf{S}} := -\boldsymbol{\theta} \odot \frac{\hat{\nabla}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta} + \mu \hat{\mathbf{g}}) - \hat{\nabla}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})}{\mu}$$

Using ZO gradient estimates $\widehat{\nabla}_{\theta} f$ to estimate Hessian-gradient product

How to Scale Up ZO Optimization in DL Training?

- **Reducing query complexity of CGE** via "pruned gradients".
  → Sparse Gradient, Dense Model. ⭐

- **Proposed technique**: Model pruning via ZO oracle

  Sparse mask via ZO gradient signal preservation (ZO-GraSP)

$$\hat{\mathbf{S}} := -\boldsymbol{\theta} \odot \frac{\hat{\nabla}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta} + \mu \hat{\mathbf{g}}) - \hat{\nabla}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})}{\mu}$$

- **Sparse-CGE that leverages layer-wise sparsity ratio**

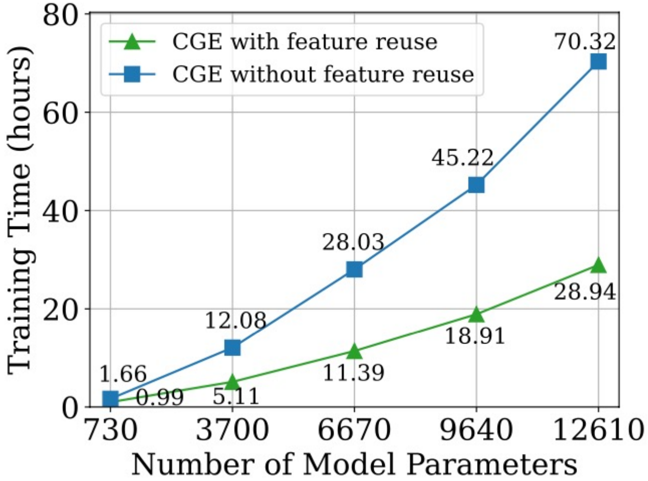
$$\hat{\nabla}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \sum_{i \in \mathcal{S}_{\text{ZO-GraSP}}} \left[\frac{\ell(\boldsymbol{\theta} + \mu \mathbf{e}_i) - \ell(\boldsymbol{\theta})}{\mu} \mathbf{e}_i \right]$$

Further Enhancing the Scalability of ZO Optimization

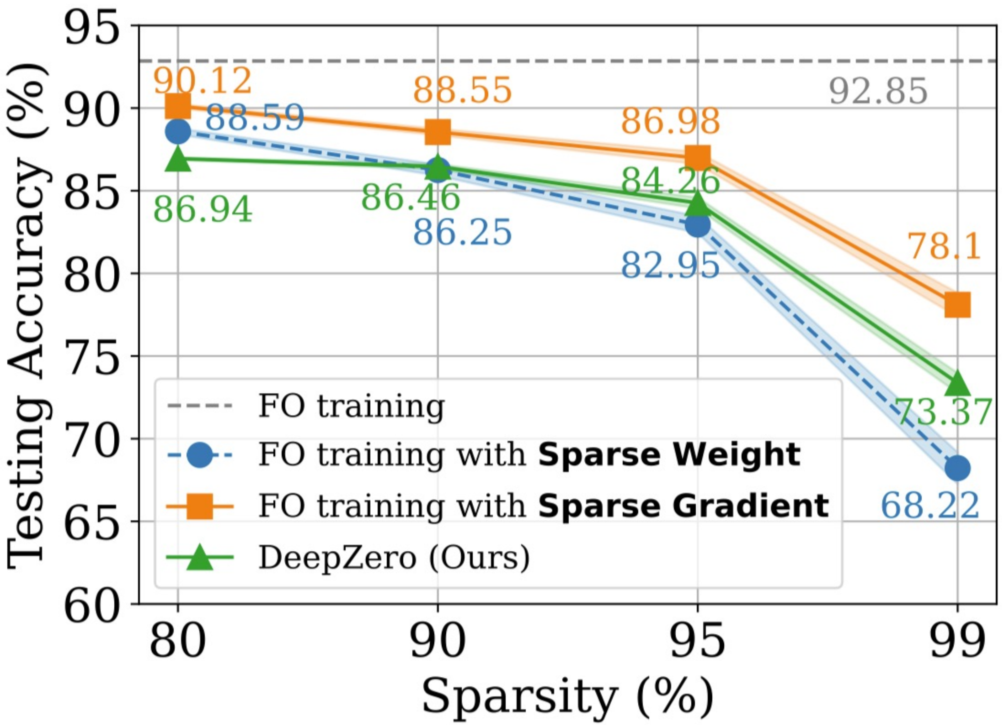
- **Parallelization of coordinate-wise finite differences**

$$\hat{\nabla}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \sum_{i=1}^{M} \hat{\mathbf{g}}_i, \quad \hat{\mathbf{g}}_i := \sum_{j \in \mathcal{S}_i} \left[\frac{\ell(\boldsymbol{\theta} + \mu \mathbf{e}_j) - \ell(\boldsymbol{\theta})}{\mu} \mathbf{e}_j \right]$$

- **Feature Reuse**: CGE perturbs each parameter element-wise. Thus, one can **reuse the feature immediately preceding the perturbed layer**

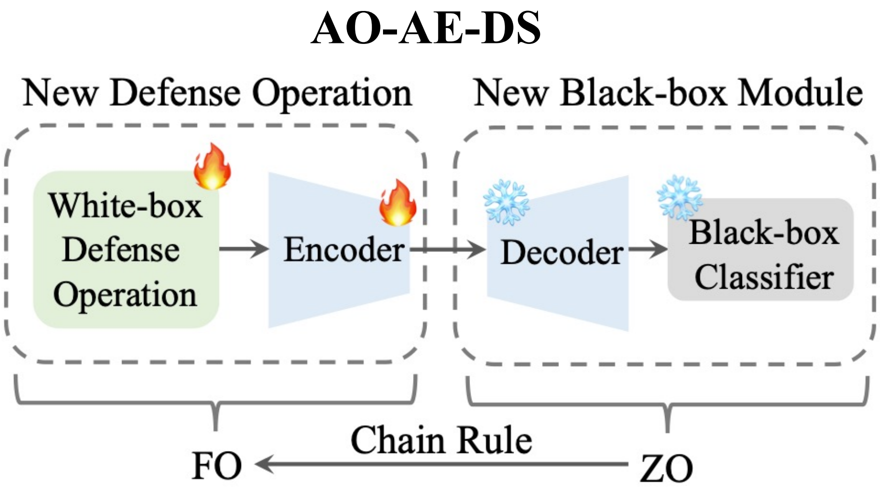


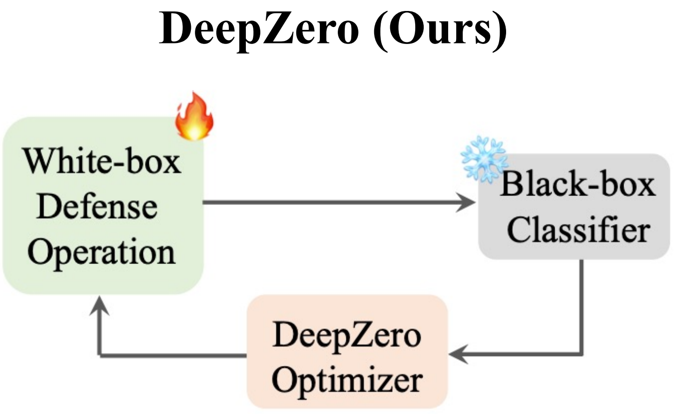
Application: Image classification



DeepZero vs. FO training on (ResNet-20, CIFAR-10)

Application: Black-box defense

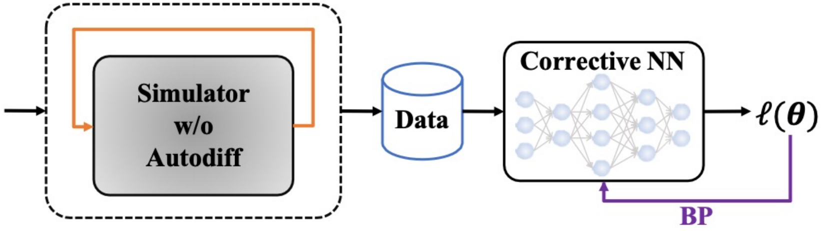




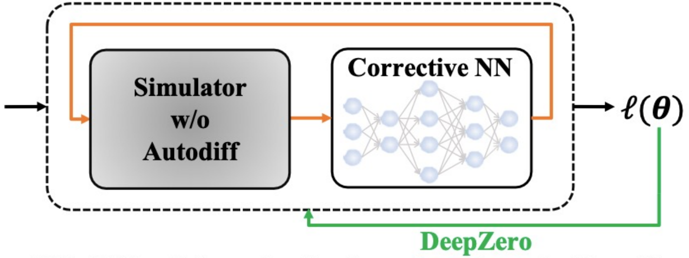
ImageNet (10 classes)			
Radius r	FO-DS	ZO-AE-DS	DeepZero
0.0	89.33	63.60	86.02
0.25	81.67	52.80	76.61
0.5	68.87	43.13	61.80
0.75	49.80	32.73	43.05

Application: Simulation-coupled DL

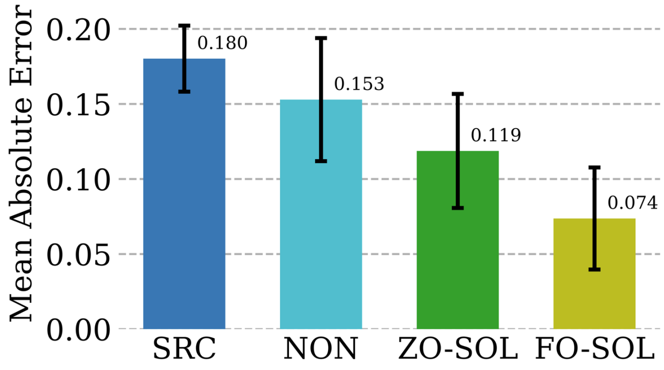
Solver-in-the loop (SOL): Training a corrective NN through looping interactions with the iterative partial differential equation (PDE) solver



NON: Non-interactive training out of the simulation loop



ZO-SOL: Solver-in-the-loop training via DeepZero



- **SRC** (low fidelity simulation without error correction)
- **NON** (non-interactive training using pre-generated low and high fidelity simulation data)

Um, et al. "Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers." NeurIPS'20

Summary

- **Scaling up ZO optimization for DL training is NON-trivial !**
- (Insight 1) **CGE outperforms RGE** in computation efficiency and accuracy
- (Insight 2) **Pruning via ZO oracle** can be used to reduce query complexity of CGE
- (Insight 3) Improved scalability can be achieved via **feature reuse** and **computing parallelization**

Scan For Code

Acknowledgement

We thank the U.S. Department of Energy via Lawrence Livermore National Laboratory under Contract DE-AC52- 07NA27344 and the LLNL-LDRD Program under Project No. 23-ER-030 (LLNL-CONF-849161) for their support.

Lawrence Livermore National Laboratory

MICHIGAN STATE UNIVERSITY

terima kasih

mulțumesc

ありがとう

谢谢 ngiyabonga suksema

Met dank
obrigada

Thank You

baie dankie
molte grazie

merci

감사합니다

Danke schön!

謝謝

obrigado

gracias

Благодарность شكرًا

Спасибі Dziękuję

dank u

mahalo

tusind tak