Machine learning for neutron data analytics

Guannan Zhang
Senior Research Staff
Data Analytics and Machine Learning Group
Computer Science and Mathematics Division
Email: zhangg@ornl.gov

Collaborators: Huibo Cao, Bryan Chakoumakos, Dan Lu, Siyan Liu, Junqi Yin, Feiyi Wang, Hoang Tran

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Scalable and uncertainty-aware neural networks for feature extraction

Publication:
J. Yin, G. Zhang, H. Cao, S. Dash, B. Chakoumakos, F. Wang, *Toward an Autonomous Workflow for Single Crystal Neutron Diffraction*, ***Smoky Mountains Computational Sciences and Engineering Conference (SMC2022)***

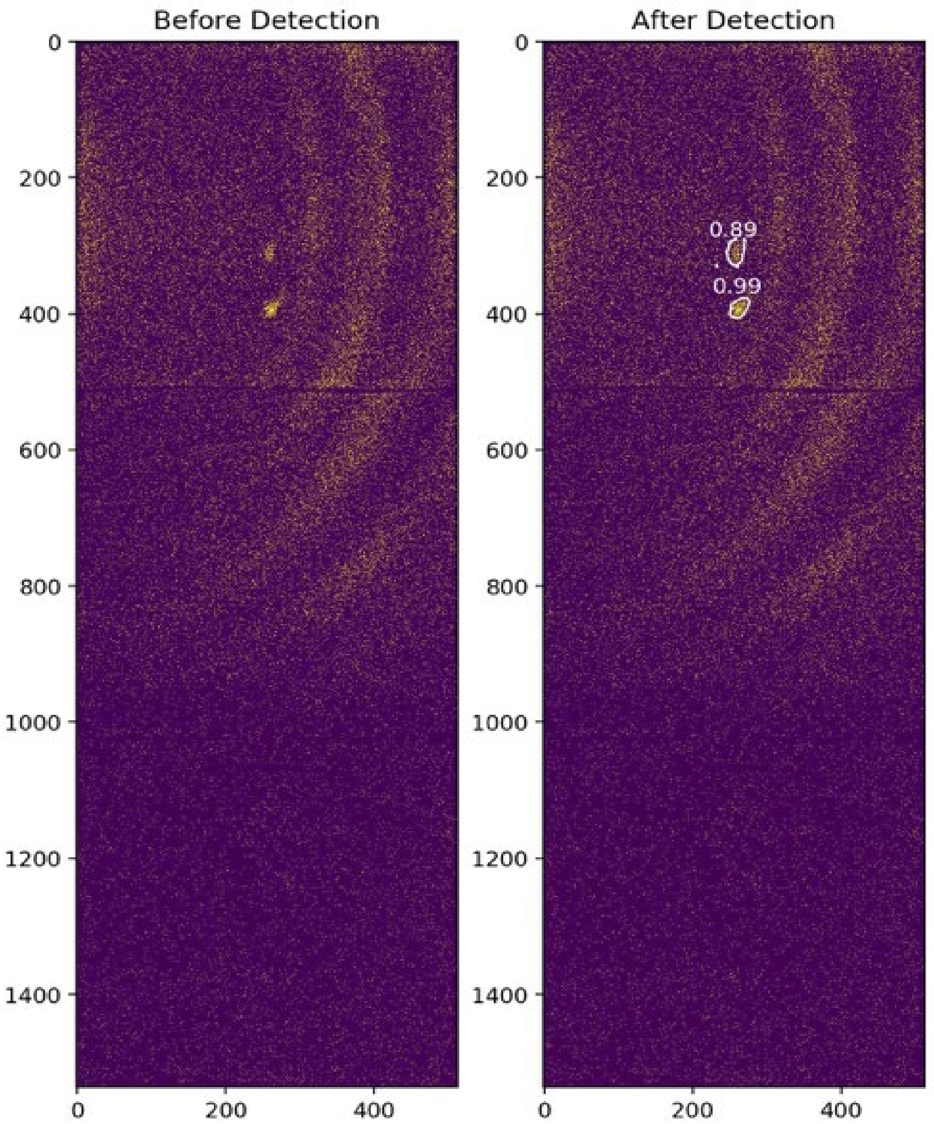
Target problem: detecting and extracting Bragg peaks

Objectives:

- Automated detection of Bragg peaks from neutron diffraction images.
- Generate a polygon mask to describe the shape of each detected peak.

Requirements:

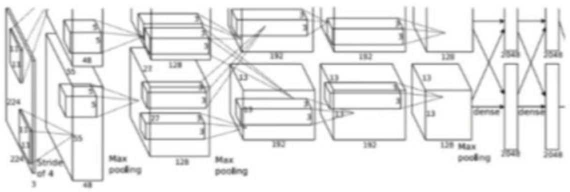
- Reliability: low false positive and false negative rates.
- Uncertainty quantification: provide a quantified confidence score for each detected Bragg peak.



Region-based convolutional neural networks (R-CNN) for object detection

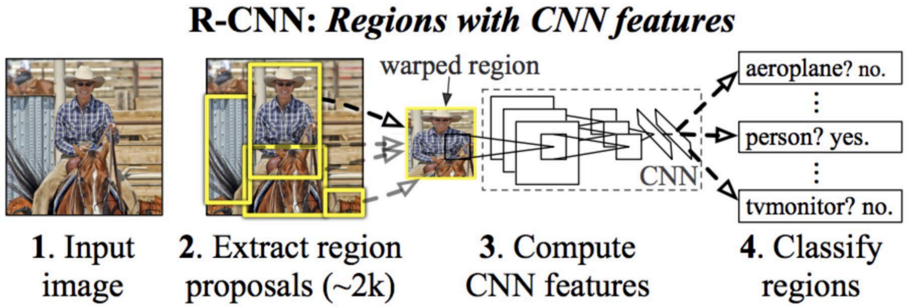
Goal: draw a bounding box around the object of interest to locate it within the image.

The **standard CNN** for classification



Drawback:
The objects of interest might have different spatial locations within the image and different aspect ratios. Hence, you would have to select a huge number of regions and this could computationally blow up.

The **R-CNN** for object detection

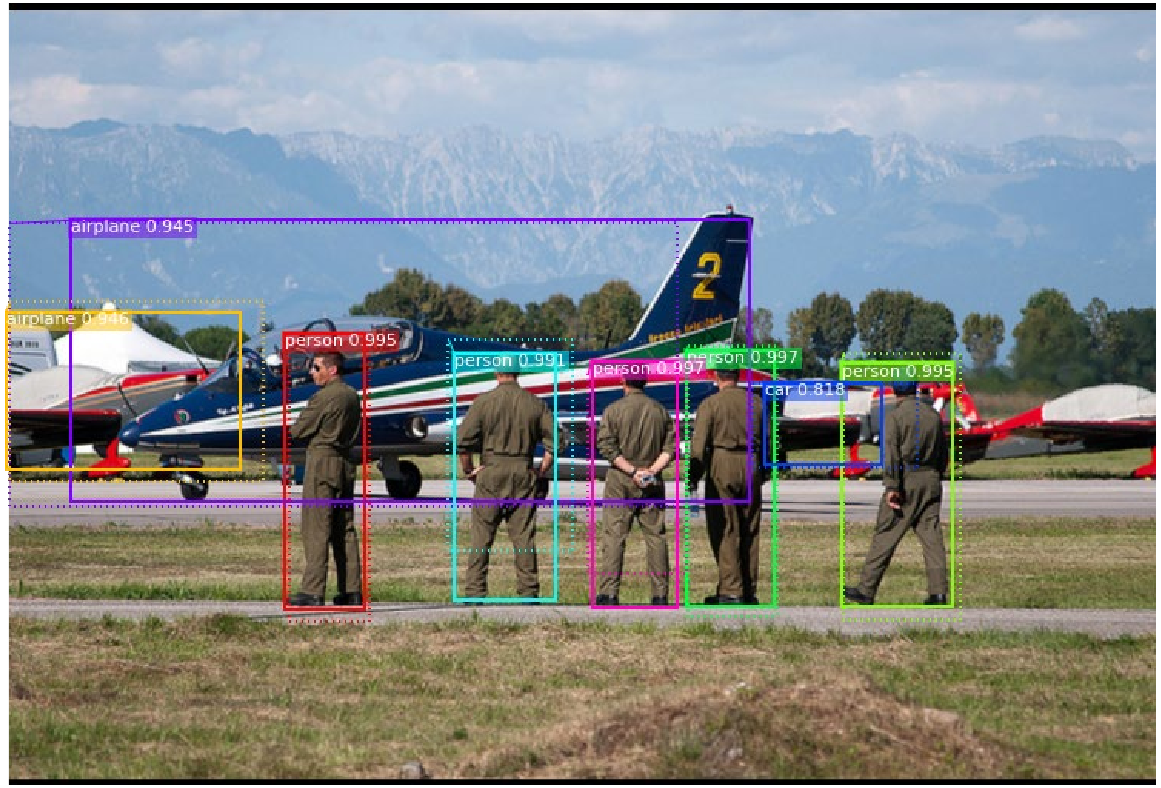


Advantage:
Use a selective search (i.e., greedy algorithm) to extract just 2000 regions from the image and he called them region proposals.

K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980-2988,

Region-based convolutional neural networks (R-CNN) for object detection

R-CNN for generating bounding boxes



Mask R-CNN for generating masks

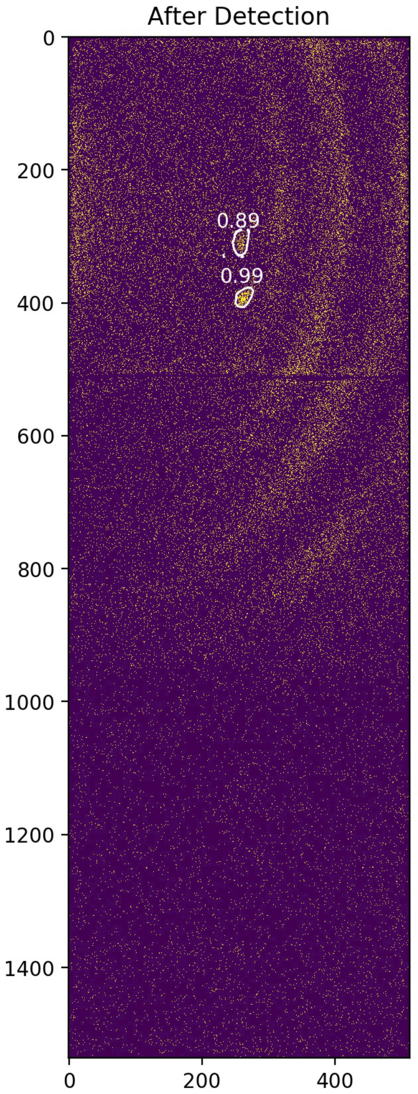
R-CNN for extracting Bragg peaks from neutron diffraction data

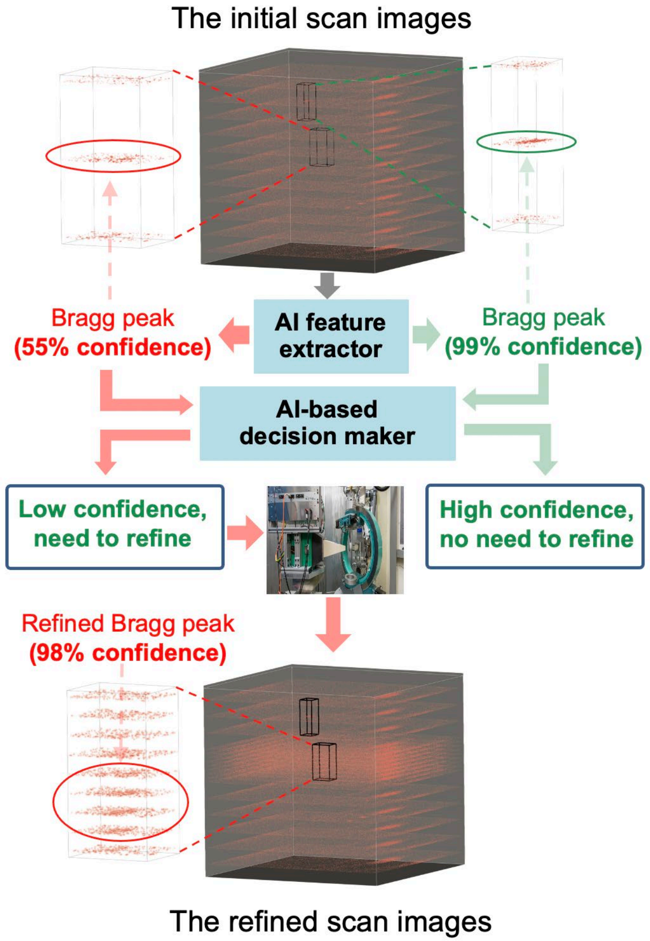
Our method:

- We used about 400 labeled data to train R-CNN model
- We used transferred learning by only training the fully-connected layers and using the pre-trained convolutional layers.

We improved the R-CNN model from two aspects:

- We developed ensemble training (i.e., training multiple R-CNNs and using the averaged model to make predictions) to improve the reliability of the confidence score.
- We developed a smoothing technique to generate masks with smoother boundaries.





A prototype Web-based app for AI-based feature extraction

- We have been implementing a prototype web service for neutron experimentalists to use our AI tools.
- The trained AI models are stored on a DGX box at OLCF.
- Experimentalists can use web browsers to access and use the tools.
- The AI model can provide real-time feedback to guide adjustment of experiment parameters.
- When updating AI models, we only need to update the backend, which is transparent to the experimentalists.

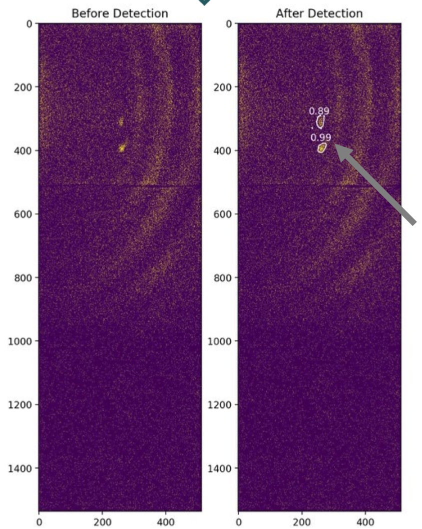
The backend at OCLF

Data Transfer

The web-based frontend

Produce AI-based analysis

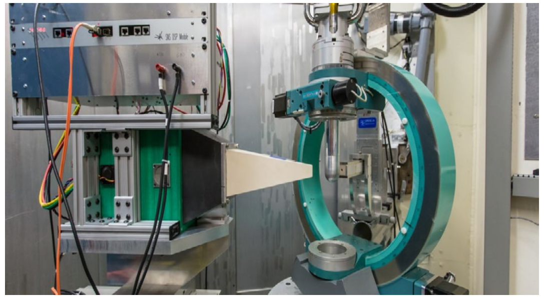
Raw Data



Identified Bragg Peaks

Use the analysis to automatically adjust experiment parameters

The neutron diffraction instrument



High-dimensional black-box optimization

Publication:
J. Zhang, H. Tran, D. Lu, and G. Zhang, *Enabling long-range exploration in minimization of multimodal functions*, ***Proceedings of 37th Conference on Uncertainty in Artificial Intelligence (UAI)***, PMLR 161: 1639-1649, 2021.

Black-box optimization

- **Problem:** Let $F : \mathbb{R}^d \to \mathbb{R}$, find $\boldsymbol{x} \in \mathbb{R}^d$ to minimize F, i.e.,

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} F(\boldsymbol{x})$$

- The Black-box scenario: we can only evaluate $F(\boldsymbol{x})$ for a given $\boldsymbol{x}$
  - The local gradient $\nabla F(\boldsymbol{x})$ is unavailable
  - The local gradient is not very useful

Non-smooth, noisy

- Typical examples:
  - A complex legacy simulator is involved in $F(\boldsymbol{x})$, e.g., parameter estimation.
  - Optimizing neural network architecture and hyper-parameters.
  - Reinforcement learning.
  - Generating adversarial examples for training robust AI models.

Gaussian smoothing for local gradient estimation

Gaussian approximation of $F(\boldsymbol{x})$ [Flaxman et al., 2005; Nesterov, Spokoiny, 2015]

$$F_\sigma(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} F(\boldsymbol{x} + \sigma \boldsymbol{u})\, \mathrm{e}^{-\frac{1}{2}\|\boldsymbol{u}\|_2^2}\, d\boldsymbol{u}$$
$$= \mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(0,\mathbf{I}_d)} \left[F(\boldsymbol{x} + \sigma \boldsymbol{u})\right],$$

Gradient of $F_\sigma(\boldsymbol{x})$ is given by

$$\nabla F_\sigma(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}\sigma} \int_{\mathbb{R}^d} F(\boldsymbol{x} + \sigma \boldsymbol{u})\boldsymbol{u}\, \mathrm{e}^{-\frac{1}{2}\|\boldsymbol{u}\|_2^2}\, d\boldsymbol{u},$$
$$= \frac{1}{\sigma}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(0,\mathbf{I}_d)} \left[F(\boldsymbol{x} + \sigma \boldsymbol{u})\,\boldsymbol{u}\right].$$

Algorithm (simple random search): Iteration $t \geq 0$.

1. Generate $\boldsymbol{u} \sim \mathcal{N}(0, \mathbf{I}_d)$ and compute $F(\boldsymbol{x}_t + \sigma\boldsymbol{u})\,\boldsymbol{u}$.
2. Compute $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \dfrac{\lambda}{\sigma} F(\boldsymbol{x}_t + \sigma\boldsymbol{u})\boldsymbol{u}$,

where λ is the learning rate.

Challenges and our objectives

Challenges:

1. High dimensionality
   - Monte Carlo methods lead to inaccurate gradient estimation, because the expectation $\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(0,\mathbf{I}_d)}[\cdot]$ is a d-dimensional integral.
2. Complex landscape of objective functions
   - Most existing works focus on improving the estimate of the local gradient.
   - An optimizer guided by the local gradient is often trapped in local optima when minimizing multi-modal functions.

Objectives:

1. Develop a new Gaussian smoothing method that only involves very low-dimensional integrals.
2. Develop a nonlocal gradient that can perform long-range exploration to avoid local minima in optimizing multi-modal functions.
3. Develop accurate estimator of the nonlocal gradient.

Directional Gaussian Smoothing

The key idea:

- Conduct 1D long-range explorations along d orthogonal directions in $\mathbb{R}^d$, each of which defines a nonlocal directional derivative as a 1D integral.
- The Gauss-Hermite quadrature is used to estimate the d 1D integrals to provide accurate estimation of the DGS gradient.

Define a *one-dimensional* function

$$G(y \,|\, \boldsymbol{x}, \boldsymbol{\xi}) = F(\boldsymbol{x} + y\,\boldsymbol{\xi}), \quad y \in \mathbb{R}.$$

Define the Gaussian smoothing of $G(y)$, denoted by $G_\sigma(y)$, by

$$G_\sigma(y \,|\, \boldsymbol{x}, \boldsymbol{\xi}) := \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} G(y + \sigma v \,|\, \boldsymbol{x}, \boldsymbol{\xi})\, \mathrm{e}^{-\frac{v^2}{2}}\, dv = \mathbb{E}_{v \sim \mathcal{N}(0,1)} \left[G(y + \sigma v \,|\, \boldsymbol{x}, \boldsymbol{\xi}) \right].$$

The derivative of $G_\sigma(y|\boldsymbol{x}, \boldsymbol{\xi})$ at $y = 0$ is represented by

$$\mathscr{D}[G_\sigma(0 \,|\, \boldsymbol{x}, \boldsymbol{\xi})] = \frac{1}{\sqrt{2\pi}\sigma} \int_{\mathbb{R}} G(\sigma v \,|\, \boldsymbol{x}, \boldsymbol{\xi})\, v\, \mathrm{e}^{-\frac{v^2}{2}}\, dv = \frac{1}{\sigma} \mathbb{E}_{v \sim \mathcal{N}(0,1)} \left[G(\sigma v \,|\, \boldsymbol{x}, \boldsymbol{\xi})\, v \right],$$

where $\mathscr{D}$ denotes the differential operator.

An Illustration of our DGS gradient

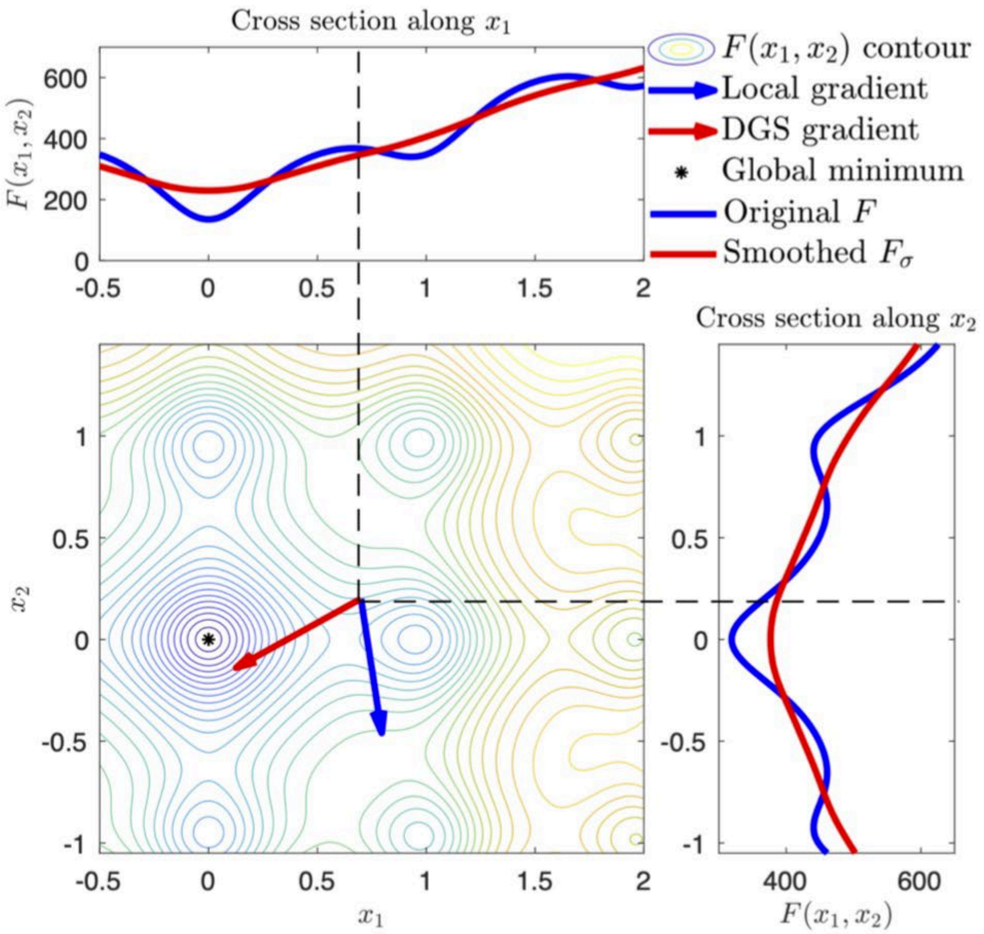
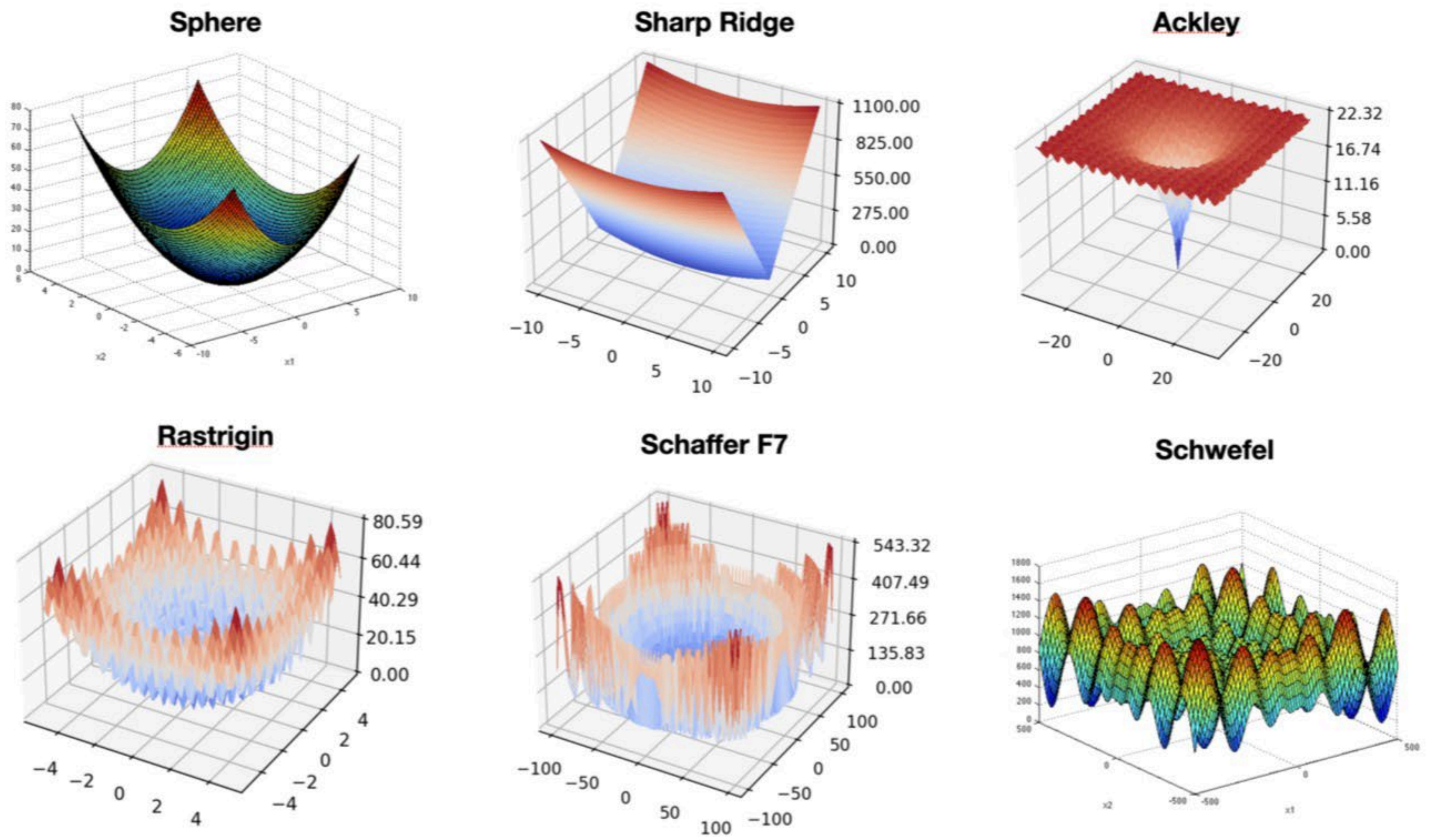


Illustration of the nonlocal exploration of the DGS gradient. In the central plot, the blue arrow points to the *local* gradient direction and the red arrow points to the DGS gradient direction. The top and right plots show the directionally smoothed functions along the two axes. Because the DGS gradient captures the major structure of F along both directions, it provides a direction pointing much closer to the global minimum than the local gradient.

Tests on benchmark global optimization problems

We show the performance of our method using the following benchmark functions. More tests can be found on `https://github.com/HoangATran/AdaDGS`.



Comparison with the state-of-the-art methods in 2000D dim

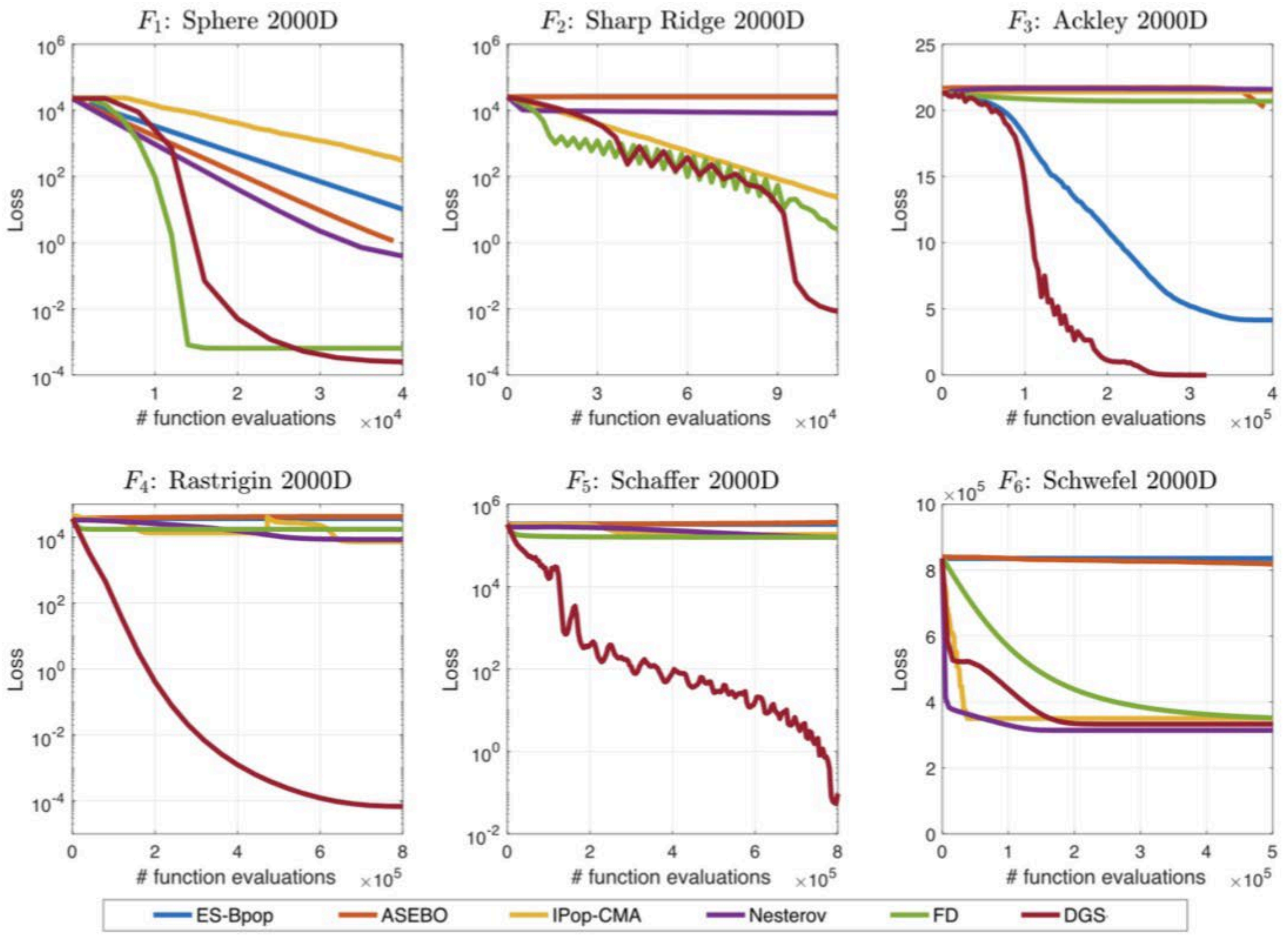


Figure: ES-Bpop: classic MC-based gradient estimation, ASEBO: MC-based gradient estimation with dimension reduction, IPop-CMA: CMA-ES method with random restarts, Nesterov: the Nesterov's random search, FD: finite difference, DGS: our method

Thank you for your attention!