

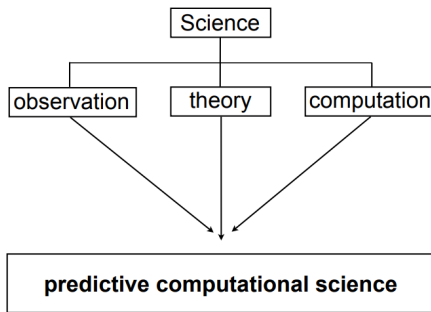
# DAS: A deep adaptive sampling method for solving partial differential equations

Peng Cheng Laboratory

Kejun Tang  
tangkj@pcl.ac.cn

Joint work with Xiaoliang Wan (Louisiana State University) and Chao Yang (Peking University)

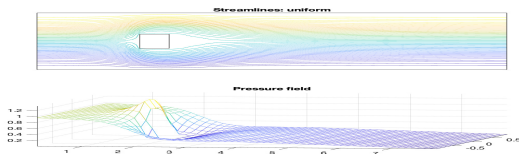
# Background



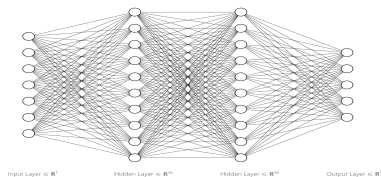
- Electromagnetic field
- Aerospace
- Molecular dynamics
- ...

# Background

- Mathematical (physical) model: PDEs or ODEs

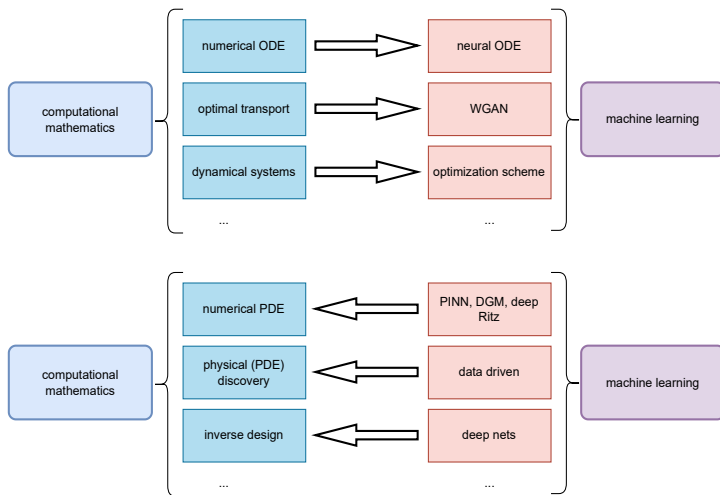


- Data-driven model (e.g., deep neural networks): no proper physical model but massive available data



- Numerical methods  
Both of them need numerical methods

# Machine learning & scientific computing (Scientific machine learning)



# Machine learning & scientific computing (Scientific machine learning)

- Uncertainty quantification (UQ): (Bayesian) Surrogate model, [Zhu and Zabaras, 2018]; Physical informed neural networks [Raissi, Perdikaris and Karniadakis, 2018]
- Density estimation and sampling method:  
Neural importance sampling, [Müller et.al, 2019]; Flow model for model reduction, [Wan and Wei, 2020]; Neural ODE, [Chen et.al, 2018]; Real NVP, [Dinh, Sohl-Dickstein and Bengio, 2016]; GAN, [Goodfellow et. al, 2014]; VAE, [Kingma and Welling, 2014]
- Deep neural networks for PDEs:  
Deep Ritz, [E and Yu, 2017] ; PDE-Net, [Long et. al, 2018]; PINN for PDE [Raissi, Perdikaris and Karniadakis, 2019]; Deep Galerkin [Sirignano and Spiliopoulos, 2018]; Physical constraint, [Zhu and Zabaras, 2019]; D3M, [Li, Tang, Wu and Liao, 2019]; PFNN, [Sheng and Yang, 2020]

# Goal

## Traditional numerical method

- high fidelity
- suffers from the curse of dimensionality

## Machine (deep) learning approach

- low fidelity
- weaker dependence on dimensionality

our purpose:

## Develop a deep adaptive method for solving PDEs

- deep networks to alleviate curse of dimensionality
- develop **adaptive** scheme using machine learning technique

# Deep learning for PDE

$$\begin{aligned}\mathcal{L}(x; u(x)) &= s(x) & \forall x \in \Omega, \\ \mathfrak{b}(x; u(x)) &= g(x) & \forall x \in \partial\Omega.\end{aligned}$$

$\mathcal{L}$  : partial differential operator,  $\mathfrak{b}$  : boundary operator.

**FEM:**

1. **mesh**
2. **basis**



**Deep methods:**

1. **samples**
2. **neural networks**

## Why deep methods

- fast inference
- attack high dimensional problems

# Deep learning for PDE

$$\begin{aligned}\mathcal{L}(x; u(x)) &= s(x) & \forall (x) \in \Omega, \\ \mathfrak{b}(x; u(x)) &= g(x) & \forall (x) \in \partial\Omega.\end{aligned}$$

$\mathcal{L}$  : partial differential operator,  $\mathfrak{b}$  : boundary operator.

How deep methods do: a deep nets  $u(x; \Theta) \rightarrow u(x)$

$$J(u(x; \Theta)) = \|r(x; \Theta)\|_{2, \Omega}^2 + \gamma \|b(x; \Theta)\|_{2, \partial\Omega}^2,$$

where  $r(x; \Theta) = \mathcal{L}u(x; \Theta) - s(x)$ ,  $b(x; \Theta) = \mathfrak{b}u(x; \Theta) - g(x)$ , and

$$\|r(x; \Theta)\|_{2, \Omega}^2 = \int_{\Omega} r^2(x; \Theta) dx$$

An optimization problem:  $\min_{\Theta} J(u(x; \Theta))$



# Deep learning for PDE

$$\begin{aligned}\mathcal{L}(x; u(x)) &= s(x) & \forall (x, ) \in \Omega, \\ \mathfrak{b}(x; ; u(x, )) &= g(x) & \forall (x, ) \in \partial\Omega.\end{aligned}$$

$\mathcal{L}$  : partial differential operator,  $\mathfrak{b}$  : boundary operator.

How deep methods do: a deep nets  $u(x; \Theta) \rightarrow u(x)$

$$J_N(u(x; \Theta)) = \frac{1}{N_r} \sum_{i=1}^{N_r} r^2(x_{\Omega}^{(i)}; \Theta) + \hat{\gamma} \frac{1}{N_b} \sum_{i=1}^{N_b} b^2(x_{\partial\Omega}^{(i)}; \Theta),$$

$x_{\Omega}^{(i)}$  drawn from  $\Omega$  and  $x_{\partial\Omega}^{(i)}$  drawn from  $\partial\Omega$

Key point:  $\min_{\Theta} J(u(x; \Theta)) \rightarrow \min_{\Theta} J_N(u(x; \Theta))$  discretize loss by **uniform sampling**

# Deep learning for PDE

$$u(x; \Theta^*) = \arg \min_{\Theta} J(u(x; \Theta)),$$

$$u(x; \Theta_N^*) = \arg \min_{\Theta} J_N(u(x; \Theta)).$$

$$\mathbb{E} (\|u(x; \Theta_N^*) - u(x)\|_{\Omega}) \leq \underbrace{\mathbb{E} (\|u(x, \Theta_N^*) - u(x; \Theta^*)\|_{\Omega})}_{\text{statistical error}} + \underbrace{\|u(x; \Theta^*) - u(x)\|_{\Omega}}_{\text{approximation error}}$$

Our work: focus on how to reduce the statistical error

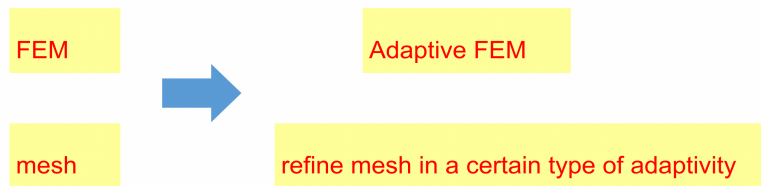
the capability of neural networks  $\rightarrow$  approximation error

the strategy of loss discretization  $\rightarrow$  statistical error

Key point: how to sample?

# Adaptivity

Question: is uniform sampling optimal for deep method?



Observation: uniform mesh is not optimal for FEM

Deep method

lack of adaptivity → develop adaptive scheme

# Adaptivity

- How does FEM do?

## Error estimator

mathematical framework: using error estimator to refine mesh

- How does deep method do?

???

we need a general mathematical framework...

# Deep adaptive sampling method (DAS)

## How deep methods do: a viewpoint of variance reduction

$$J_r(u(x; \Theta)) = \int_{\Omega} r^2(x; \Theta) dx = \int_{\Omega} \frac{r^2(x; \Theta)}{p(x)} p(x) dx \approx \frac{1}{N_r} \sum_{i=1}^{N_r} \frac{r^2(x_{\Omega}^{(i)}; \Theta)}{p(x_{\Omega}^{(i)})},$$

where  $\{x_{\Omega}^{(i)}\}_{i=1}^{N_r}$  from  $p(x)$  instead of a uniform distribution.

## or relax the definition of $J_r(u)$

$$J_{r,p}(u(x; \Theta)) = \int_{\Omega} r^2(x; \Theta) p(x) dx \approx \frac{1}{N_r} \sum_{i=1}^{N_r} r^2(x_{\Omega}^{(i)}; \Theta),$$

## Importance sampling

$$p(x) = \frac{r^2(x; \Theta)}{\mu}, \quad \mu = \int_{\Omega} r^2(x; \Theta) dx$$

# Deep adaptive sampling method (DAS)

Sample from  $p(x)$  for a fixed  $\Theta$ : a deep generative model

$$p_{KRnet}(x; \Theta_f) \approx \mu^{-1} r^2(x; \Theta)$$

where  $p_{KRnet}(x; \Theta_f)$  is a PDF induced by KRnet [Tang, Wan and Liao, 2020]; [Tang, Wan and Liao, 2021]

“Error estimator”:  $\hat{r}_X(x) \propto r^2(x; \Theta)$

$$D_{KL}(\hat{r}_X(x) \| p_{KRnet}(x; \Theta_f)) = \int_B \hat{r}_X \log \hat{r}_X dx - \int_B \hat{r}_X \log p_{KRnet} dx.$$

$$\min_{\Theta_f} H(\hat{r}_X, p_{KRnet}) = - \int_B \hat{r}_X \log p_{KRnet} dx.$$

## Challenge

- design a valid PDF model for efficient sampling

# Deep adaptive sampling method (DAS)

## Deep generative models

- GAN [Goodfellow et.al, 2014] [Arjovsky, Chintala and Bottou, 2017]
  - VAE [Kingma and Welling, 2014]
  - NICE [Dinh, Krueger and Bengio, 2014], Real NVP [Dinh, Dickstein, and Bengio, 2016]
- 
- GAN & VAE generate sample efficiently
  - cannot get PDF

# Deep adaptive sampling method (DAS)

KRnet: construct a PDF model via Knothe-Rosenblatt rearrangement, [Tang, Wan and Liao, 2021]

$$z = f_{KRnet}(x) = L_N \circ f_{[K-1]}^{outer} \circ \dots \circ f_{[1]}^{outer}(x),$$

$$p_{KRnet}(x) = p_Z(f_{KRnet}(x)) |\det \nabla_x f_{KRnet}|,$$

where  $f_{[i]}^{outer}$  is defined as

$$f_{[k]}^{outer} = L_S \circ f_{[k,L]}^{inner} \circ \dots \circ f_{[k,1]}^{inner} \circ L_R.$$

## Advantages

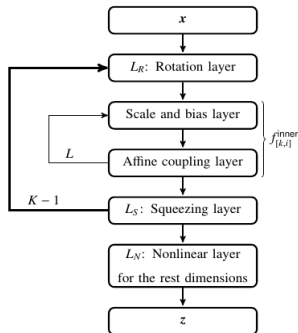
- GAN and VAE can not provide an explicit PDF though they can generate samples efficiently
- KRnet provides an explicit PDF
- KRnet can generate samples efficiently



# Deep adaptive sampling method (DAS)

structure of KRnet

- squeezing layer
- rotation layer
- affine coupling layer
- nonlinear layer



# Deep adaptive sampling method (DAS)

The framework of DAS (see [Tang, Wan and Yang, 2021] for more details)  
1

// solve PDE

Sample  $m$  samples  $x_{\Omega,k}^{(i)}$  and Sample  $m$  samples  $x_{\partial\Omega,k}^{(j)}$ .

Update  $u(x; \Theta)$  by descending the stochastic gradient of  $J_N(u(x; \Theta))$ .

// Train KRnet

Sample  $m$  samples from  $x_{\Omega,k}^{(i)}$ .

Update  $p_{KRnet}(x; \Theta_f)$  by descending the stochastic gradient of  $H(\hat{r}_X, \hat{p}_{KRnet})$ .

// Refine training set

Generate  $x_{\Omega,k+1}^{(i)} \subset \Omega$  through  $p_{KRnet}(x; \Theta_f^{*,(k+1)})$ .

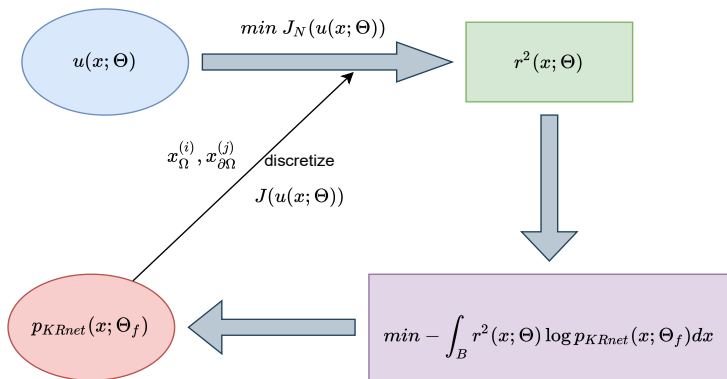
Repeat until stopping criterion satisfies

---

<sup>1</sup>K. Tang, X. Wan and C. Yang, DAS: A deep adaptive sampling method for solving partial differential equations, arXiv preprint arXiv:2112.14038, (2021).

# Deep adaptive sampling method (DAS)

The framework of DAS. (see [Tang, Wan and Yang, 2021] for more details)<sup>2</sup>



<sup>2</sup>K. Tang, X. Wan and C. Yang, DAS: A deep adaptive sampling method for solving partial differential equations, arXiv preprint arXiv:2112.14038, (2021).

# Analysis of DAS

## Theorem (Tang, Wan and Yang, 2021; informal)

Let  $u(x; \Theta_N^{*,(k)}) \in F$  be a solution of DAS at the  $k$ -stage where the collocation points are independently drawn from  $\hat{p}_{KRnet}(x; \Theta_f^{*,(k-1)})$ . Given  $0 < \varepsilon < 1$ , the following error estimate holds under certain conditions

$$\left\| u(x; \Theta_N^{*,(k)}) - u(x) \right\|_{2,\Omega} \leq \sqrt{2} C_1^{-1} \left( R_k + \varepsilon + \left\| b(x; \Theta_N^{*,(k)}) \right\|_{2,\partial\Omega}^2 \right)^{\frac{1}{2}}.$$

with probability at least  $1 - \exp(-2N_r \varepsilon^2 / (\tau_2 - \tau_1)^2)$ .

## Corollary (Tang, Wan and Yang, 2021; informal)

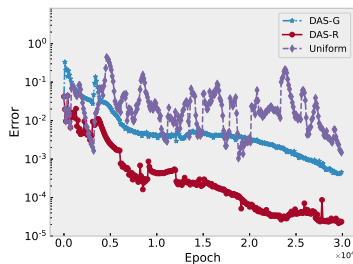
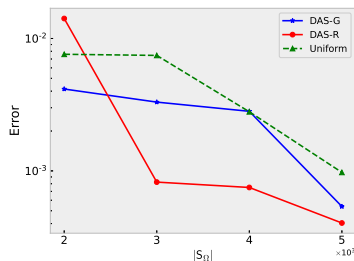
If the boundary loss  $J_b(u)$  is zero, then the following inequality holds

$$\mathbb{E}(R_{k+1}) \leq \mathbb{E}(R_k)$$

# Numerical experiments

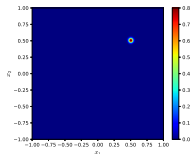
## Two-dimensional peak problem

$$\begin{aligned} -\Delta u(x_1, x_2) &= s(x_1, x_2) \quad \text{in } \Omega, \\ u(x_1, x_2) &= g(x_1, x_2) \quad \text{on } \partial\Omega, \end{aligned}$$

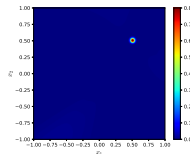


# Numerical experiments

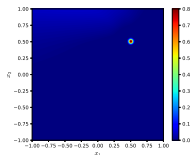
## Two-dimensional peak problem



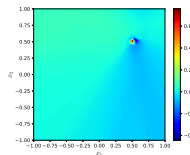
(a) The exact solution.



(b) DAS-R approximation.



(c) DAS-G approximation.

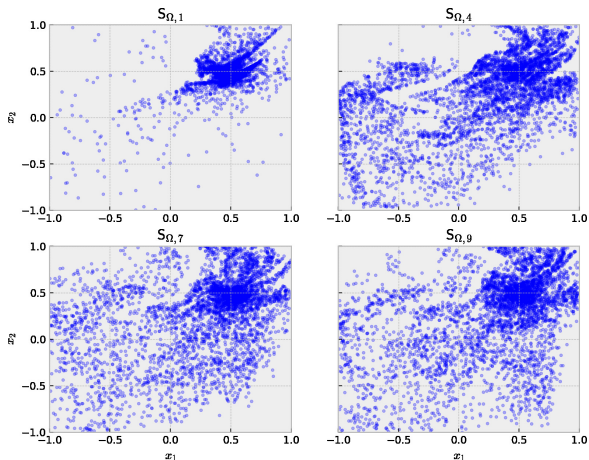


(d) Uniform sampling strategy.

# Numerical experiments

Two-dimensional peak problem

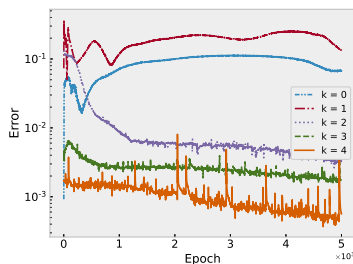
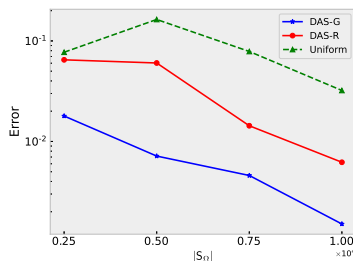
Samples



# Numerical experiments

Two-dimensional problem with two peaks

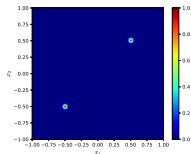
$$\begin{aligned} -\nabla \cdot [u(x_1, x_2) \nabla (x_1^2 + x_2^2)] + \nabla^2 u(x_1, x_2) &= s(x_1, x_2) \quad \text{in } \Omega, \\ u(x_1, x_2) &= g(x_1, x_2) \quad \text{on } \partial\Omega, \end{aligned}$$



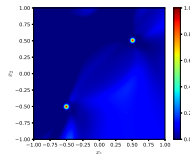


# Numerical experiments

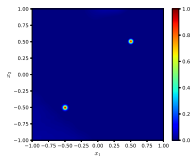
Two-dimensional problem with two peaks



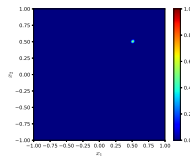
(e) The exact solution.



(f) DAS-R approximation.



(g) DAS-G approximation.

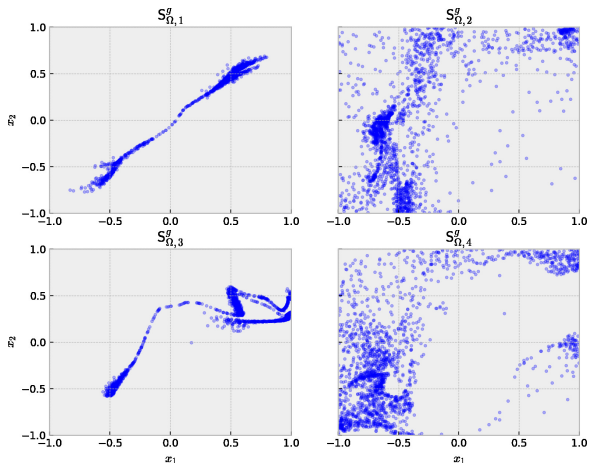


(h) Uniform sampling strategy.

# Numerical experiments

Two-dimensional problem with two peaks

Samples



# Numerical experiments

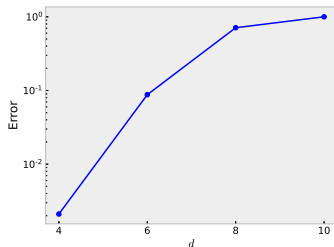
The  $d$ -dimensional elliptic equation

$$-\Delta u(x) = s(x), \quad x \text{ in } \Omega = [-1, 1]^d,$$

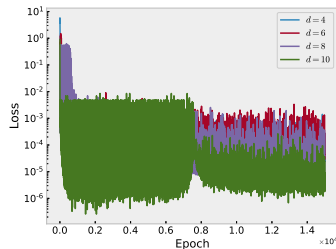
with an exact solution

$$u(x) = e^{-10\|x\|_2^2},$$

where the Dirichlet boundary condition on  $\partial\Omega$  is given by the exact solution.



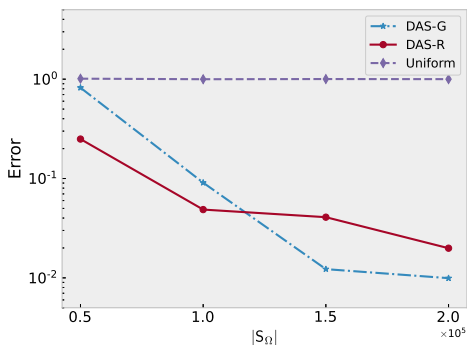
(i) Error



(j) Loss

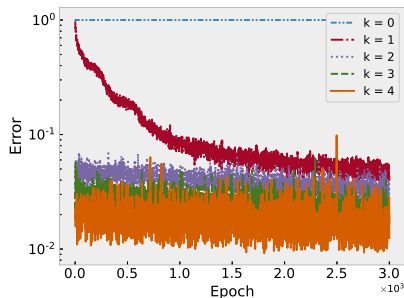
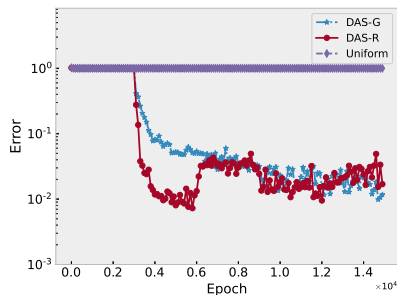
# Numerical experiments

The 10-dimensional elliptic equation



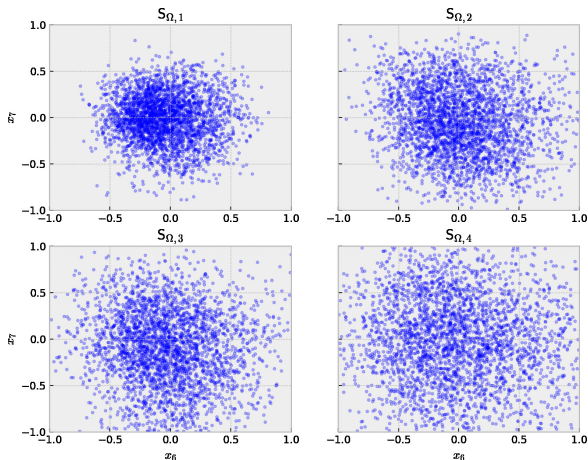
# Numerical experiments

## The 10-dimensional elliptic equation



# Numerical experiments

The 10-dimensional elliptic equation  
Samples



# Summary of DAS

## summary

- significantly improve the accuracy for PDEs with low regularity problems especially when the dimensionality is relatively large
- a very general and flexible framework for the adaptive learning strategy

## outlook

- time-dependent problems
- real-world applications

# Thank you for your attention