Deep adaptive sampling for surrogate modeling: Algorithm, Theory, and Applications

PKU-Changsha Institute for Computing and Digital Economy

Kejun Tang
joint work with Qifeng Liao, Xiaoliang Wan, Xili Wang, Jiayu Zhai and Chao Yang

April 20, 2024

ShanghaiTech University

Outline

Big data era: data-driven

Using data to train a predictive model with parameters Θ

$$u(\mathbf{x}; \Theta)$$

e.g. deep neural networks

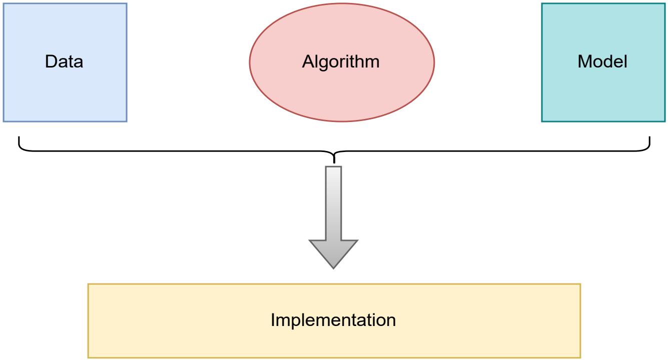
Training usually means an optimization problem

$$\text{unsupervised} \quad \min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} J(x^{(i)}; \Theta) \qquad \text{supervised} \quad \min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} J(x^{(i)}, y^{(i)}; \Theta).$$

where J is a proper loss function, e.g. mean square error, cross entropy etc.

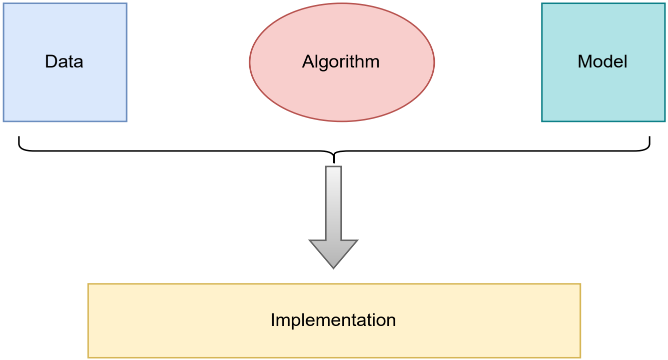
- machine learning
- computer vision
- signal processing
- ...

Big data era: data-driven



- Model: deep neural networks, physical model, or coupling
- Data: labeled, unlabled, random samples ....
- Algorithm: various optimization methods

Big data era: data-driven



data is oil

- model is driven by data
- data has the influence on generalization

Goal

Traditional numerical methods

- high fidelity
- suffers from the curse of dimensionality

Machine (deep) learning approaches

- low fidelity
- weaker dependence on dimensionality

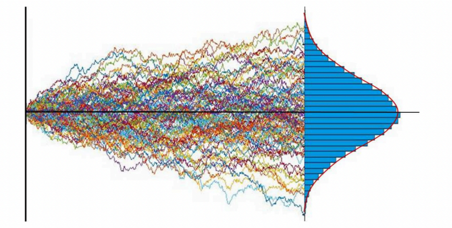
our purpose:

Develop adaptive numerical methods by data-driven modes for high-dimensional or low-regularity problems

- deep networks to alleviate the curse of dimensionality
- develop adaptive schemes for the machine learning solver

Examples

High-dimensional problems, e.g., Fokker-Planck equations



[Image courtesy of M. Mohammadi]

Low-regularity problems, e.g., the lid-driven cavity problem

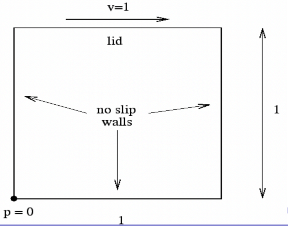


Illustration of the statistical error

A function approximation perspective

Let $\mathbf{X} \in \mathbb{R}^d$ and $Y \in \mathbb{R}$ subject to a joint distribution $\rho_{\mathbf{X},\mathbf{Y}}$
$\hat{Y} = m(\mathbf{X})$: a model
$h : \mathbf{x} \mapsto y$ a function to be approximated
We know in the L_2 sense the optimal model is

$$m^* = \arg\min_m \left[L(m) = \int (y - m(\mathbf{x}))^2 \rho_{\mathbf{X},Y}(\mathbf{x}, y) d\mathbf{x} dy \right].$$

$$m_{\mathbf{w}^*} = \arg\min_{m_{\mathbf{w}} \in W} \left[L_N(m_{\mathbf{w}}) = \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - m_{\mathbf{w}}(\mathbf{x}^{(i)}))^2 \right],$$

L_N: a Monte Carlo approximation of L with dataset $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$

Illustration of the statistical error

A function approximation perspective

A linear space $V = \mathrm{span}\{q_i : i = 1, \dots, n\}$

$$m_{\hat{\mathbf{v}}^*} = \arg\min_{m_{\hat{\mathbf{v}}} \in V} \left[L_{V,N}(m_{\hat{\mathbf{v}}}) = \frac{1}{N} \sum_{i=1}^{N} \left(m_{\hat{\mathbf{v}}}(\mathbf{x}^{(i)}) - h(\mathbf{x}^{(i)}) \right)^2 \right],$$

Lemma (Tang, Wan and Yang, 2022)

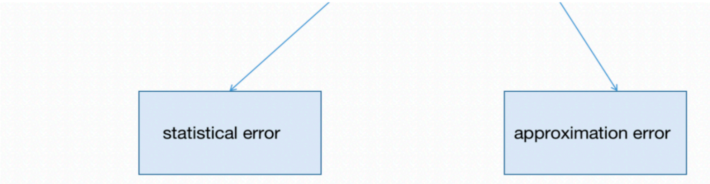
Let $h \in C(D)$ be a continuous function defined on a compact domain $D \subset \mathbb{R}^d$ and $\rho(\mathbf{x}) > 0$ be a PDF on D. Let $V = \mathrm{span}\{q_i : i = 1, \dots, n\}$ with q_i being orthonormal polynomials in terms of ρ. For any $\delta > 0$ and with probability at least $1 - 2\delta$, we have for a sufficiently large N

$$\|m_{\hat{\mathbf{v}}^*} - h\|_\rho \leq C \sqrt{\frac{\ln \delta^{-1}}{N}} + \|m_V^* - h\|_\rho,$$

where C is a constant, and $\|\cdot\|_\rho$ is the weighted L_2 norm in terms of ρ.

Illustration of the statistical error

$$\|m_{\hat{\mathbf{v}}^*} - h\|_\rho \leq C\sqrt{\frac{\ln \delta^{-1}}{N}} + \|m_V^* - h\|_\rho,$$



- the hypothesis space $V \to$ approximation error
- the training set $\to$ statistical error

Partial differential equations

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

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

FEM:

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

Deep methods:

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

Why deep methods

- fast inference
- tackle high dimensional problems

Deep learning for PDEs

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

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

How deep methods do: a deep net $u(\mathbf{x};\Theta) \to u(\mathbf{x})$

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

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

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

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

Deep learning for PDEs

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

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

How deep methods do: a deep net $u(\mathbf{x}; \Theta) \to u(\mathbf{x})$

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

$\mathbf{x}_{\Omega}^{(i)}$ drawn from Ω and $\mathbf{x}_{\partial\Omega}^{(i)}$ drawn from $\partial\Omega$

Key point: $\min\limits_{\Theta} J\left(u(\mathbf{x}; \Theta)\right) \to \min\limits_{\Theta} J_N\left(u(\mathbf{x}; \Theta)\right)$ discretize the loss by uniform sampling in general (or other quasi-random methods based on uniform samples)

Deep learning for PDEs

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

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

Our work: focus on how to reduce the statistical error

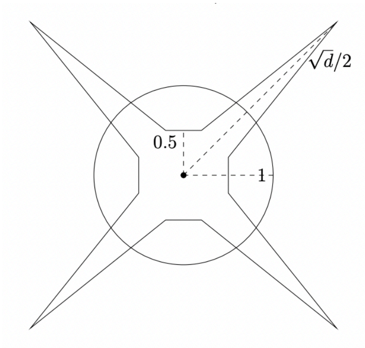
the capability of neural networks $\rightarrow$ approximation error

the strategy of loss discretization $\rightarrow$ statistical error

Key point: how to sample?

Geometric properties of high-dimensional spaces

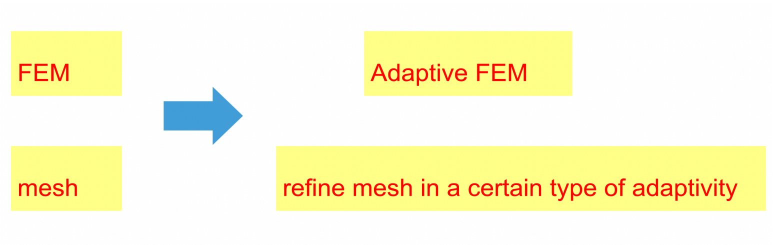
uniformly distributed points in high-dimensional spaces



Most of the volume of a high-dimensional cube is located around its corner [Vershynin, High-Dimensional Probability, 2020]. Cube: $[-1, 1]^d$

$$\mathbb{P}(\|\mathbf{x}\|_2^2 \leq 1) \leq \exp(-\frac{d}{10}).$$

Question: is uniform sampling optimal for deep methods?



Observation:

1. uniform mesh is not optimal for FEM
2. choosing uniform samples is not a good choice for high-dimensional problems

Deep methods

lack of adaptivity $\rightarrow$ develop adaptive schemes

Localized residual

Assume

$$\zeta = \int_\Omega 1_I(\mathbf{x})d\mathbf{x} \approx \int_\Omega r^2(\mathbf{x})d\mathbf{x} \ll 1.$$

A rare event!

Consider a Monte Carlo estimator of ζ in terms of uniform samples

$$\hat{P}_{\mathrm{MC}} = \frac{1}{N}\sum_{i=1}^{N} 1_I(\mathbf{x}^{(i)}).$$

The relative error of $\hat{P}_{\mathrm{MC}}$ is

$$\frac{\mathrm{Var}^{1/2}(\hat{P}_{\mathrm{MC}})}{\zeta} = N^{-1/2}((1-\zeta)/\zeta)^{1/2} \approx (\zeta N)^{-1/2}.$$

sample size $O(1/\zeta) \to$ relative error $O(1)$.

Adaptivity

- How does FEM do?

Error estimator

general framework: using an error estimator to refine mesh

- How does deep method do?

???

we need a general framework...

Deep adaptive sampling method (DAS)

How deep methods do: a viewpoint of variance reduction

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

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

or relax the definition of $J_r(u)$

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

Importance sampling

$$p^* = \frac{r^2(\mathbf{x};\Theta)}{\mu},\ \mu = \int_\Omega r^2(\mathbf{x};\Theta)d\mathbf{x}$$

Deep adaptive sampling method (DAS)

Sample from $p(\mathbf{x})$ for a fixed Θ: a deep generative model

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

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

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

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

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

Challenge

- design a valid PDF model for efficient sampling

Deep adaptive sampling method (DAS)

Lemma (Tang, Wan and Yang, 2022)

Assume that $|\Omega| = 1$ *and* $p(\mathbf{x})$ *is a PDF satisfying*

$$D_{\mathrm{KL}}(p \| p^*) \leq \varepsilon < \infty.$$

For any $0 < a < \infty$, *we have*

$$\mathbb{E}\left|Q_p[r^2] - \mathbb{E}[r^2]\right| \leq a N_r^{-1/2} + 2\|r^2/p\|_p \sqrt{\mathbb{P}(|r^2/p - \mu| > a; p)},$$

where

$$Q_p(r^2) = \frac{1}{N_r}\sum_{i=1}^{N_r} \frac{r^2(\mathbf{X}^{(i)})}{p(\mathbf{X}^{(i)})}, \mathbf{X}^{(i)} \sim p(\mathbf{x}),$$

and

$$\mathbb{P}(|r^2/p - \mu| > a; p) \leq \frac{\mu(2\varepsilon)^{1/2}}{a}.$$

Key ingredient of DAS: deep generative models with an explicit PDF

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

Key ingredient of DAS: deep generative models with an explicit PDF

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

$$\mathbf{z} = f_{KRnet}(\mathbf{x}) = L_N \circ f^{\text{outer}}_{[K-1]} \circ \cdots \circ f^{\text{outer}}_{[1]}(\mathbf{x}),$$

$$p_{KRnet}(\mathbf{x}) = p_{\mathbf{Z}}(f_{KRnet}(\mathbf{x})) \left| \det \nabla_{\mathbf{x}} f_{KRnet} \right|,$$

where $f^{\text{outer}}_{[i]}$ is defined as

$$f^{\text{outer}}_{[k]} = L_S \circ f^{\text{inner}}_{[k,L]} \circ \cdots \circ f^{\text{inner}}_{[k,1]} \circ L_R.$$

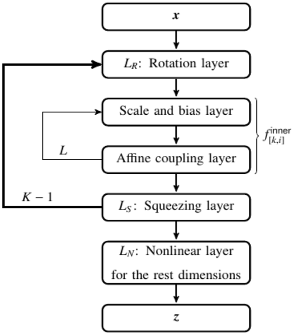
Advantages

- 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

Key ingredient of DAS: deep generative models with an explicit PDF

structure of KRnet

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



$$\mathbf{z} = \mathcal{T}^{-1}(\mathbf{x}) = \begin{bmatrix} \mathcal{T}_1(x_1) \\ \mathcal{T}_2(x_1, x_2) \\ \vdots \\ \mathcal{T}_N(x_1, \dots, x_N) \end{bmatrix}$$

An affine coupling layer

Each $f_{[i]}$

- $f_{[i]}$ is a bijection
- $\det \nabla_{\mathbf{x}} f_{[i]}$ can be easily computed
- $|\det \nabla_{\mathbf{x}} f| = \prod_{i=1}^{L} |\det \nabla_{\mathbf{x}_{[i-1]}} f_{[i]}|$

structure of $f_{[i]}$

$$\mathbf{x}_{[i],1} = \mathbf{x}_{[i-1],1}$$
$$\mathbf{x}_{[i],2} = \mathbf{x}_{[i-1],2} \odot \left(1 + \alpha \tanh(\mathbf{s}_i(\mathbf{x}_{[i-1],1}))\right) + e^{\beta_i} \odot \tanh(\mathbf{t}_i(\mathbf{x}_{[i-1],1})),$$

where $\mathbf{x}_{[i]} = [\mathbf{x}_{[i],1}, \mathbf{x}_{[i],2}]^{\mathsf{T}} \in \mathbb{R}^d$, $\mathbf{s}_i : \mathbb{R}^m \mapsto \mathbb{R}^{d-m}$ and $\mathbf{t}_i : \mathbb{R}^m \mapsto \mathbb{R}^{d-m}$ are the scaling and the translation depending on $\mathbf{x}_{[i-1],1}$

$$(\mathbf{s}_i, \mathbf{t}_i) = \mathrm{NN}_{[i]}(\mathbf{x}_{[i-1],1}).$$

An affine coupling layer

Each $f_{[i]}$

- $f_{[i]}$ is a bijection
- $\det \nabla_{\mathbf{x}} f_{[i]}$ can be easily computed
- $|\det \nabla_{\mathbf{x}} f| = \prod_{i=1}^{L} |\det \nabla_{\mathbf{x}_{[i-1]}} f_{[i]}|$

inverse and determinant of Jacobian for $f_{[i]}$

$$\mathbf{x}_{[i-1],1} = \mathbf{x}_{[i],1}$$

$$\mathbf{x}_{[i-1],2} = \left(\mathbf{x}_{[i],2} - e^{\beta_i} \odot \tanh(\mathbf{t}_i(\mathbf{x}_{[i-1],1}))\right) \odot \left(1 + \alpha \tanh(\mathbf{s}_i(\mathbf{x}_{[i-1],1}))\right)^{-1}$$

$$\nabla_{\mathbf{x}_{[i-1]}} f_{[i]} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \nabla_{\mathbf{x}_{[i-1],1}} \mathbf{x}_{[i],2} & \mathrm{diag}(1 + \alpha \tanh(\mathbf{s}_i(\mathbf{x}_{[i-1],1}))) \end{bmatrix}$$

An affine coupling layer

structure of $f_{[i]}$

$$\mathbf{x}_{[i],1} = \mathbf{x}_{[i-1],1}$$
$$\mathbf{x}_{[i],2} = \mathbf{x}_{[i-1],2} \odot \left(1 + \alpha \tanh(\mathbf{s}_i(\mathbf{x}_{[i-1],1}))\right) + e^{\beta_i} \odot \tanh(\mathbf{t}_i(\mathbf{x}_{[i-1],1})),$$

where $\mathbf{x}_{[i]} = [\mathbf{x}_{[i],1}, \mathbf{x}_{[i],2}]^{\mathsf{T}} \in \mathbb{R}^d$, $\mathbf{s}_i : \mathbb{R}^m \mapsto \mathbb{R}^{d-m}$ and $\mathbf{t}_i : \mathbb{R}^m \mapsto \mathbb{R}^{d-m}$ are the scaling and the translation depending on $\mathbf{x}_{[i-1],1}$

advantages

- adapts the trick of ResNet [He et. al, 2015]
- e^{β_i} depends on the data points directly instead of the value of $\mathbf{x}_{[i-1]}$
- $(1-\alpha)^{d-m} \leq \det\left(\nabla_{\mathbf{x}_{[i-1]}} f_{[i]}\right) \leq (1+\alpha)^{d-m}, \alpha \in (0,1)$

Algorithm of DAS

The framework of DAS (see [Tang, Wan and Yang, 2022] for more details) [1]

// solve PDE

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

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

// Train KRnet

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

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

// Refine training set (replace all points: DAS-R; the number of points increases gradually: DAS-G)

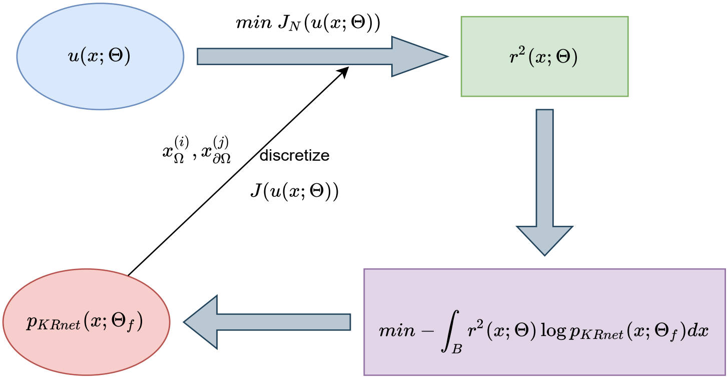
Generate $\mathbf{x}_{\Omega,k+1}^{(i)} \subset \Omega$ through $p_{KRnet}(\mathbf{x};\Theta_f^{*,(k+1)})$.

Repeat until stopping criterion satisfies

[1] K. Tang, X. Wan and C. Yang, DAS: A deep adaptive sampling method for solving partial differential equations, arXiv preprint arXiv:2112.14038, (2022).

Algorithm of DAS

The framework of DAS. (see [Tang, Wan and Yang, 2022] for more details) [2] code: `https://github.com/MJfadeaway/DAS`



[2] K. Tang, X. Wan and C. Yang, DAS: A deep adaptive sampling method for solving partial differential equations, arXiv preprint arXiv:2112.14038, (2022).

Analysis of DAS

Theorem (Tang, Wan and Yang, 2022)

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

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

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

Corollary (Tang, Wan and Yang, 2022)

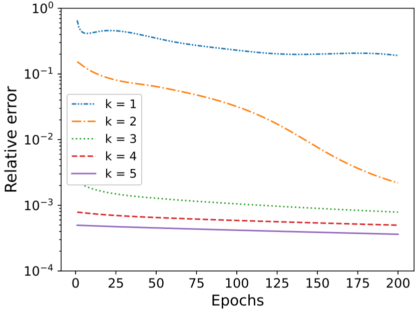
If the boundary loss $J_b(u)$ is zero, then the following inequality holds

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

Fokker-Planck equations

A special case:

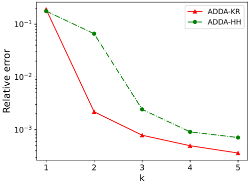
$u(\mathbf{x}) = p(\mathbf{x})$



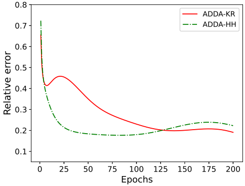
setting

- $\frac{\partial p(\mathbf{x},t)}{\partial t} = \nabla \cdot [p(\mathbf{x},t)\nabla \log(\beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}))] + \nabla^2 p(\mathbf{x},t)$
- stationary solution
  $p_{st}(\mathbf{x}) = \beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}), \mathbf{x} \in \mathbb{R}^2$, $p_i(\mathbf{x})$: Gaussian distribution

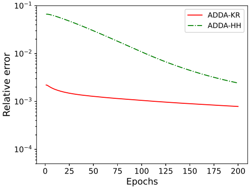
Fokker-Planck equations



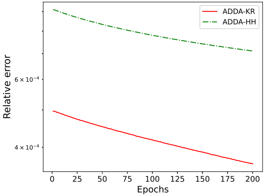
(a) KL divergece w.r.t. k-th model.



(b) The convergence behavior for $k = 1$.

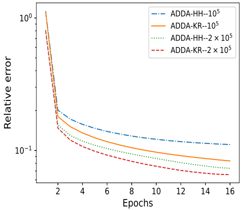


(c) The convergence behavior for $k = 3$.

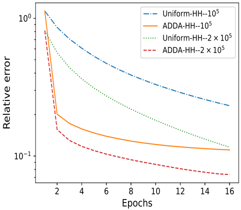


(d) The convergence behavior for $k = 5$.

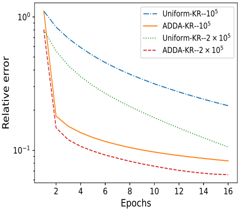
Fokker-Planck equations



(e) Comparison of KR and HH: KL-divergence w.r.t epochs



(f) KL-divergence w.r.t epochs for HH

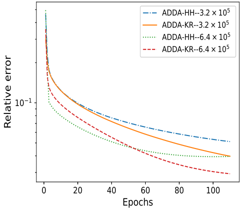


(g) KL-divergence w.r.t epochs for KR

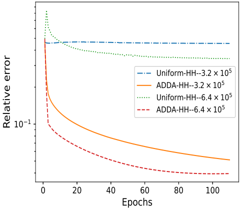
setting (HH refers to Real NVP)

- $\frac{\partial p(\mathbf{x},t)}{\partial t} = \nabla \cdot [p(\mathbf{x},t)\nabla \log(\beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}))] + \nabla^2 p(\mathbf{x},t)$
- stationary solution
  $p_{st}(\mathbf{x}) = \beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}), \mathbf{x} \in \mathbb{R}^4$, $p_i(\mathbf{x})$: Gaussian distribution

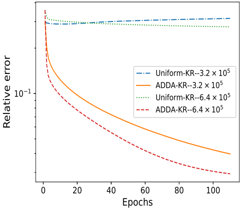
Fokker-Planck equations



(h) Comparison of KR and HH: KL-divergence w.r.t epochs



(i) KL-divergence w.r.t epochs for HH



(j) KL-divergence w.r.t epochs for KR

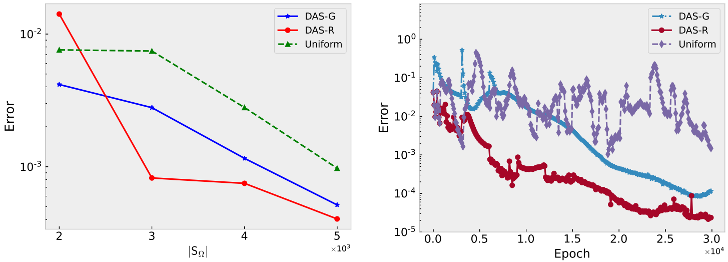
setting

- $\frac{\partial p(\mathbf{x},t)}{\partial t} = \nabla \cdot [p(\mathbf{x},t)\nabla \log(\beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}))] + \nabla^2 p(\mathbf{x},t)$
- stationary solution
  $p_{st}(\mathbf{x}) = \beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}), \mathbf{x} \in \mathbb{R}^8, p_i(\mathbf{x})$: Gaussian distribution

Elliptic PDEs: low-dimensional and low regularity cases

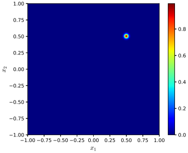
Two-dimensional peak problem

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

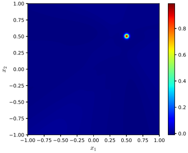


Elliptic PDEs: low-dimensional and low-regularity cases

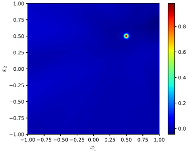
Two-dimensional peak problem



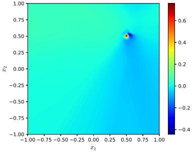
(k) The exact solution.



(l) DAS-R approximation.



(m) DAS-G approximation.

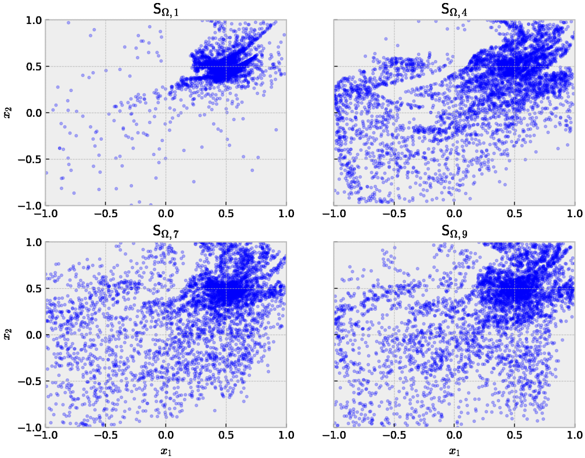


(n) Uniform sampling strategy.

Elliptic PDEs: low-dimensional and low-regularity cases

Two-dimensional peak problem

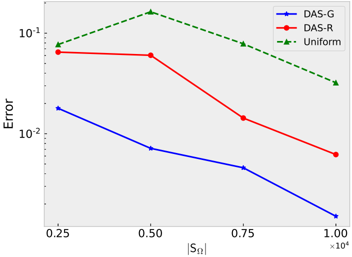
DAS-R samples

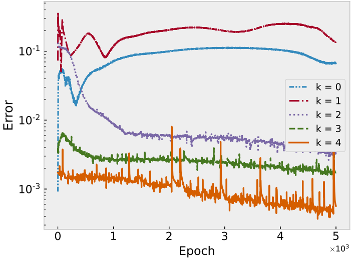


Elliptic PDEs: low-dimensional and low-regularity cases

Two-dimensional problem with two peaks

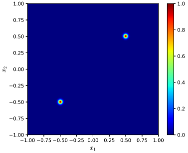
$$-\nabla \cdot \left[u(x_1, x_2)\nabla(x_1^2 + x_2^2)\right] + \nabla^2 u(x_1, x_2) = s(x_1, x_2) \quad \text{in } \Omega,$$
$$u(x_1, x_2) = g(x_1, x_2) \quad \text{on } \partial\Omega,$$



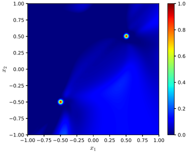


Elliptic PDEs: low-dimensional and low-regularity cases

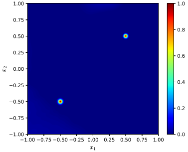
Two-dimensional problem with two peaks



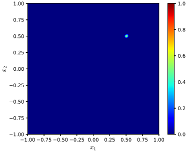
(o) The exact solution.



(p) DAS-R approximation.



(q) DAS-G approximation.

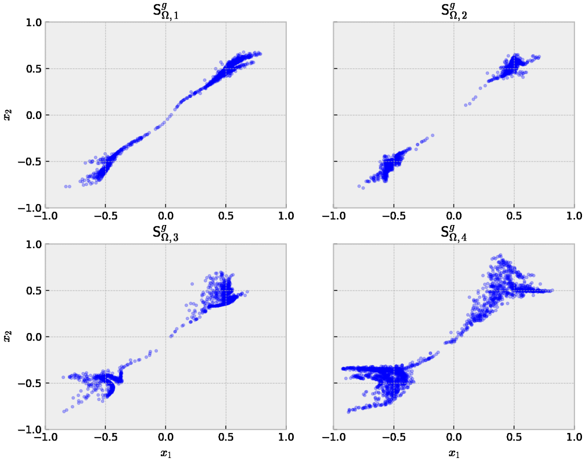


(r) Uniform sampling strategy.

Elliptic PDEs: low-dimensional and low-regularity cases

Two-dimensional problem with two peaks

DAS-G samples



Linear PDEs: High-dimensional and low-regularity cases

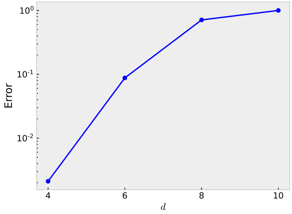
The d-dimensional linear equation

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

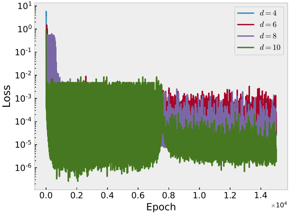
with an exact solution

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

where the Dirichlet boundary condition on $\partial\Omega$ is given by the exact solution. The uniform sampling method becomes less effective as d increases



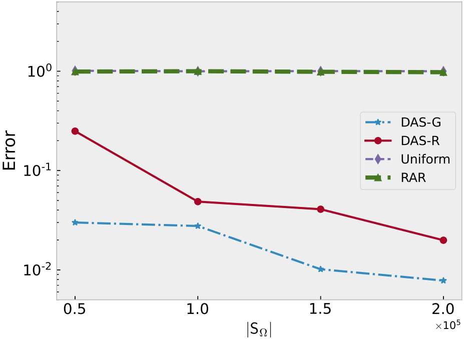
(s) Error



(t) Loss

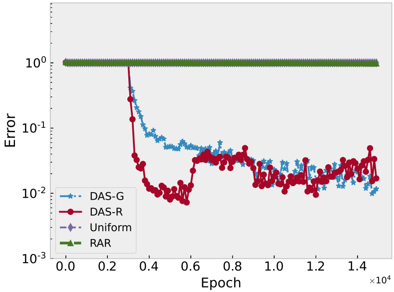
Linear PDEs: High-dimensional and low-regularity cases

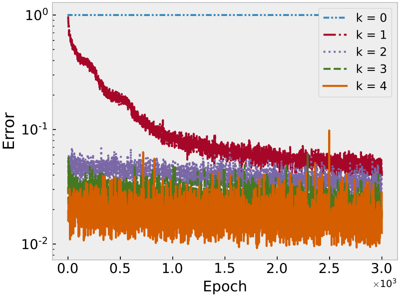
The 10-dimensional linear equation



Linear PDEs: High-dimensional and low-regularity cases

The 10-dimensional linear equation

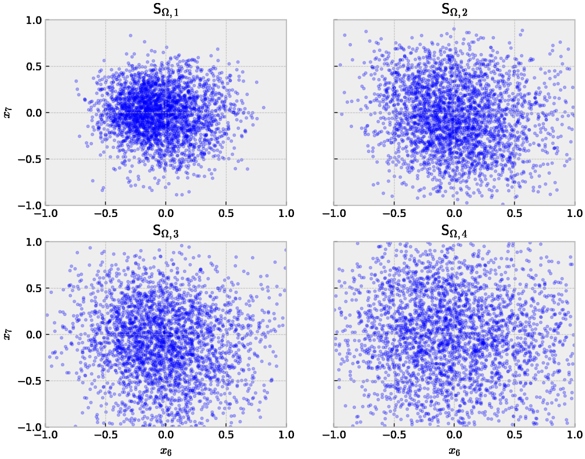




Linear PDEs: High-dimensional and low-regularity cases

The 10-dimensional linear equation

DAS-R samples



Linear PDEs: High-dimensional and low-regularity cases

The 10-dimensional linear equation
The evolution for the variance of residual

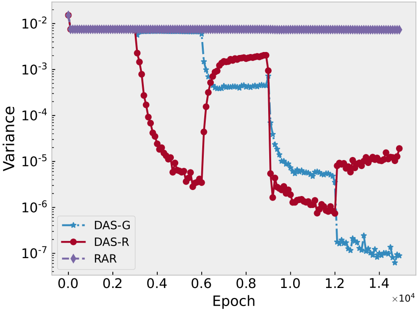


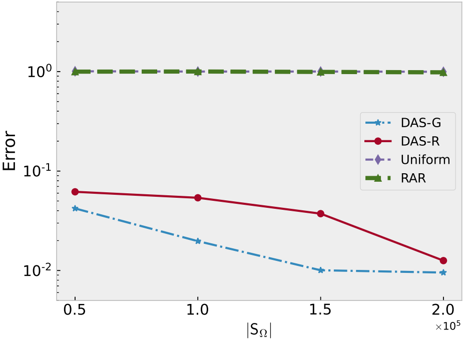
Table: Training time different $|S_\Omega|$ and sampling strategies, ten-dimensional linear test problem

$\lvert S_\Omega \rvert$	DAS-G	DAS-R	Uniform	RAR [Lu et.al, 2020]
5×10^4	1.83h	3.38h	1.90h	1.45h
10^5	3.64h	6.95h	3.92h	3.03h
1.5×10^5	5.61h	10.29h	5.85h	4.66h
2×10^5	7.55h	13.49h	7.90h	5.74h

Nonlinear PDEs: High-dimensional and low-regularity cases

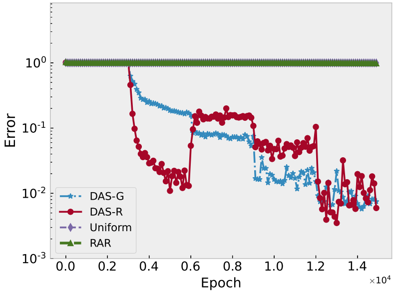
The 10-dimensional nonlinear equation

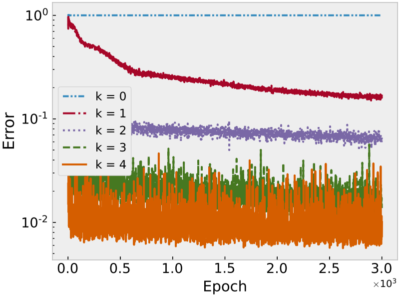
$$-\Delta u(\mathbf{x}) + u(\mathbf{x}) - u^3(\mathbf{x}) = s(\mathbf{x}), \quad \mathbf{x} \text{ in } \Omega = [-1, 1]^{10}.$$



Nonlinear PDEs: High-dimensional and low-regularity cases

The 10-dimensional nonlinear equation

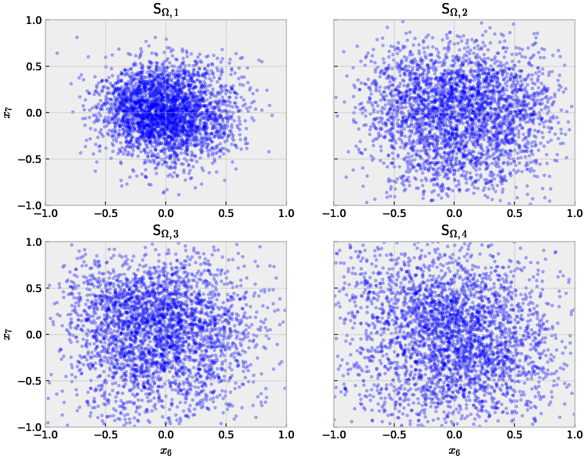




Nonlinear PDEs: High-dimensional and low-regularity cases

The 10-dimensional nonlinear equation

DAS-R samples



Elliptic PDEs: High-dimensional and low-regularity cases

The 10-dimensional nonlinear equation
The evolution for the variance of residual

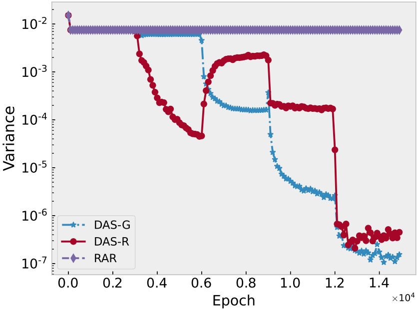


Table: Training time different $|S_\Omega|$ and sampling strategies, ten-dimensional nonlinear test problem

$\lvert S_\Omega \rvert$	DAS-G	DAS-R	Uniform	RAR [Lu et. al, 2020]
5×10^4	1.82h	3.44h	1.84h	1.42h
10^5	3.65h	6.92h	3.86h	2.97h
1.5×10^5	5.81h	10.41h	5.73h	4.63h
2×10^5	7.82h	13.87h	7.80h	5.75h

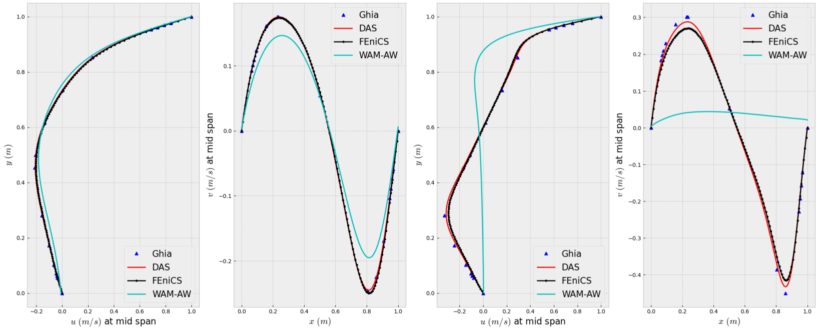
Lid-driven cavity flow problem

$$\begin{cases} \mathbf{u}(\mathbf{x}) \cdot \nabla \mathbf{u}(\mathbf{x}) + \nabla p(\mathbf{x}) = \dfrac{1}{Re} \Delta \mathbf{u}(\mathbf{x}) & \text{in } \Omega, \\ \nabla \cdot \mathbf{u}(\mathbf{x}) = 0 & \text{in } \Omega, \\ \mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) & \text{on } \partial\Omega, \end{cases}$$

- $\mathbf{u}(\mathbf{x}) = [u(\mathbf{x}), v(\mathbf{x})]^\mathsf{T}, \mathbf{x} = [x, y]^\mathsf{T}$
- $Re = 100, 400$
- The physical domain is $\Omega_s = [0, 1] \times [0, 1]$
- Boundary conditions

$$\mathbf{g}(\mathbf{x}) = \begin{cases} [1, 0]^\mathsf{T}, y = 1; \\ [0, 0]^\mathsf{T}, \ \text{otherwise}. \end{cases}$$

Lid-driven cavity flow problem



(u) $Re = 100$.

(v) $Re = 400$.

Figure: The velocity components at the location of mid-span lines for the deterministic lid-driven cavity flow problems, $Re = 100, 400$.

Lid-driven cavity flow problem

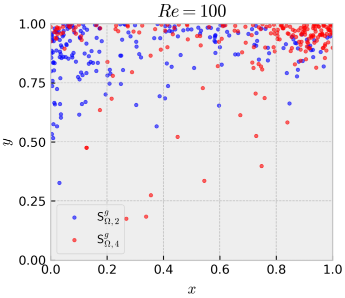
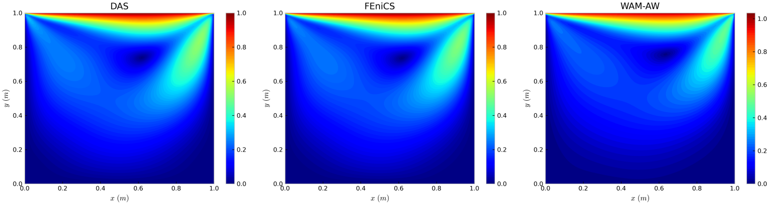
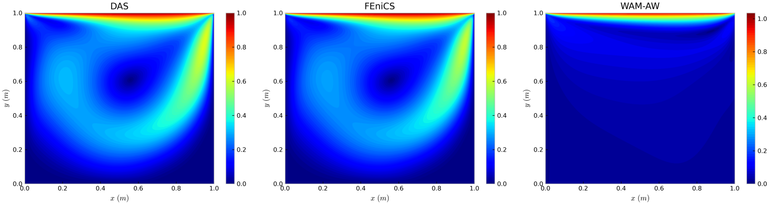


Figure: The random samples in $S^g_{\Omega,k}$ for the deterministic lid-driven cavity flow problems. Left: $S^g_{\Omega,2}$ (blue) and $S^g_{\Omega,4}$ (red) for $Re = 100$; Right: $S^g_{\Omega,2}$ (blue) and $S^g_{\Omega,9}$ (red) for $Re = 400$.

Lid-driven cavity flow problem



(a) $Re = 100$.



(b) $Re = 400$.

Prametric problems

Parametric differential equations

$$\mathcal{L}(\mathbf{x}, \xi; u(\mathbf{x}, \xi)) = s(\mathbf{x}, \xi) \qquad \forall (\mathbf{x}, \xi) \in \Omega_s \times \Omega_p,$$
$$\mathcal{B}(\mathbf{x}, \xi; u(\mathbf{x}, \xi)) = g(\mathbf{x}, \xi) \qquad \forall (\mathbf{x}, \xi) \in \partial\Omega_s \times \Omega_p.$$

For any ξ, compute the solution efficiently without solving the differential equation repeatedly.
PDF for sampler

$$p_{\mathbf{x}|\xi}(\mathbf{x}|\xi; \theta_f) = p_{\mathbf{z}|\xi}(f_{\mathrm{KRnet}}(\mathbf{x}; \xi, \theta_{\mathbf{f}})) \left|\det \nabla_{\mathbf{x}} f_{\mathrm{KRnet}}\right|.$$

$$p_{\xi|\mathbf{x}}(\xi|\mathbf{x}; \theta_f) = p_{\mathbf{z}|\mathbf{x}}(f_{\mathrm{KRnet}}(\xi; \mathbf{x}, \theta_{\mathbf{f}})) \left|\det \nabla_{\xi} f_{\mathrm{KRnet}}\right|.$$

In practice, the above two conditional PDF models can be further simplified.

Sampling strategy

- Sample from a joint PDF

$$\hat{r}(\mathbf{x}, \xi) \propto r^2(\mathbf{x}, \xi; \theta) h(\mathbf{x}, \xi),$$

- Sample from a marginal PDF

$$\tilde{r}^2(\xi; \theta) = \int_{\Omega_s} r^2(\mathbf{x}, \xi; \theta) d\mathbf{x}.$$

Analysis

Assumptions [T. De Ryck and S. Mishra, 2022]

- $\theta \in \Theta = [-a, a]^D$: trainable parameters of u_θ where $a > 0$ is a constant.
- $\mathscr{M}_1 : \theta \mapsto J_{r,N}$ and $\mathscr{M}_2 : \theta \mapsto J_r$: Lipschitz continuous in the ℓ_∞ sense with Lipschitz constant $\mathfrak{L}$ for $\theta \in \Theta$.
- Let $c > 0$ be a constant that is independent of Θ. Assume that $J_{r,N} \in [0, c]$ for all $\theta \in \Theta$.

Theorem (Wang, Tang, Zhai, Wan, and Yang, 2024)

Let θ_N^* be a minimizer of $J_{r,N}$ where the collocation points are independently drawn from a given probability distribution. Given $\varepsilon \in (0, 1)$, the following inequality holds under the above assumptions

$$J_r(u_{\theta_N^*}) \leq \varepsilon^2 + J_{r,N}(u_{\theta_N^*})$$

with probability at least $1 - (4a\mathfrak{L}/\varepsilon^2)^D \exp(-N_r\varepsilon^4/2c^2)$.

Numerical results: physics-informed operator learning

The following dynamical system

$$\begin{cases} \dfrac{\mathrm{d}u(x,\xi)}{\mathrm{d}x} = e^{-D\|\boldsymbol{\xi}-\mathbf{0.5}\|^2} f(x,\xi), \quad x \in [0,1], \\ u(0,\xi) = 0, \end{cases}$$

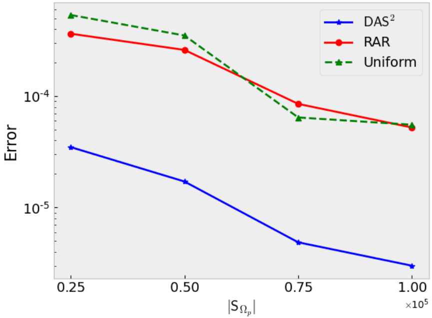
- $D = 6$: a fixed parameter
- $\xi \in \Omega_p = [-1,1]^8$

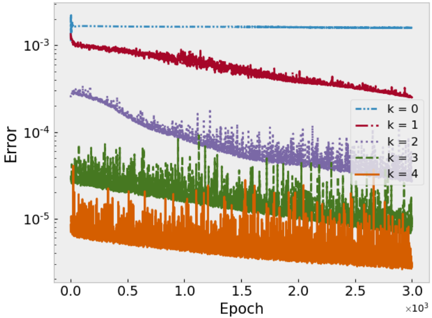
Goal: learn the sulution operator from f to the solution u without any paired input-output data

f is drawn from V_{poly} where

$$V_{\text{poly}} = \left\{ \sum_{i=0}^{d-1} \xi_i T_i(x) : |\xi_i| \le M \right\}.$$

Numerical results: physics-informed operator learning

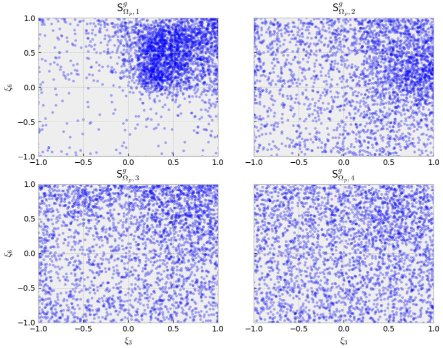




$$u_\theta(x,\xi) \approx \sum_{i=1}^{l} q_{\theta_1}^{(i)}(x)\, t_{\theta_2}^{(i)}(\xi) + b_0,$$

marginal PDF for sampling

Numerical results: physics-informed operator learning



$\lvert S_{\Omega_p}\rvert$ / sampling strategy	2.5×10^4	5×10^4	7.5×10^4	1×10^5
Uniform (0.006s)	5.4×10^{-4}	3.5×10^{-4}	6.4×10^{-5}	5.5×10^{-5}
RAR (0.006s)	3.6×10^{-4}	2.6×10^{-4}	8.5×10^{-5}	5.2×10^{-5}
DAS^2 (0.03s)	3.5×10^{-5}	1.7×10^{-5}	4.9×10^{-6}	3.0×10^{-6}

Numerical results: parametric optimal control problems

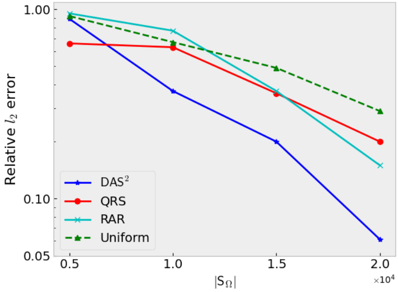
$$
\begin{cases}
\min\limits_{y(\mathbf{x},\xi),u(\mathbf{x},\xi)} J\left(y\left(\mathbf{x},\xi\right),u\left(\mathbf{x},\xi\right)\right)=\dfrac{1}{2}\left\|y(\mathbf{x},\xi)-y_d(\mathbf{x},\xi)\right\|_{2,\Omega}^2+\dfrac{\alpha}{2}\left\|u(\mathbf{x},\xi)\right\|_{2,\Omega}^2, \\
\text{subject to} \begin{cases} -\Delta y(\mathbf{x},\xi)=u(\mathbf{x},\xi) & \text{in } \Omega, \\ y(\mathbf{x},\xi)=1 & \text{on } \partial\Omega, \end{cases} \\
\text{and} \quad u_a \leq u(\mathbf{x},\xi) \leq u_b \quad \text{a.e. in } \Omega,
\end{cases}
$$

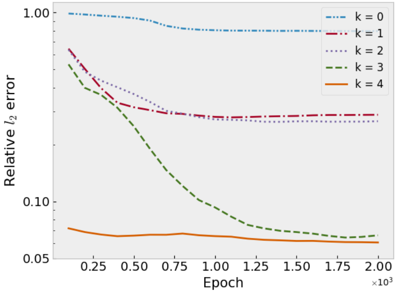
where $\Omega_p=(\xi_1,\xi_2)$ is the paramter.
$\Omega(\boldsymbol{\xi})=([0,2]\times[0,1])\backslash B((1.5,0.5),\xi_1)$ and the desired state is given by

$$
y_d(\boldsymbol{\xi})=\begin{cases} 1 & \text{in } \Omega_1=[0,1]\times[0,1], \\ \xi_2 & \text{in } \Omega_2(\boldsymbol{\xi})=([1,2]\times[0,1])\backslash B((1.5,0.5),\xi_1), \end{cases}
$$

where $B((1.5,0.5),\xi_1)$ is a ball of radius ξ_1 with center $(1.5,0.5)$, $\alpha=0.001$ and $\boldsymbol{\xi}\in\Omega_p=[0.05,0.45]\times[0.5,2.5]$.

Numerical results: parametric optimal control problems





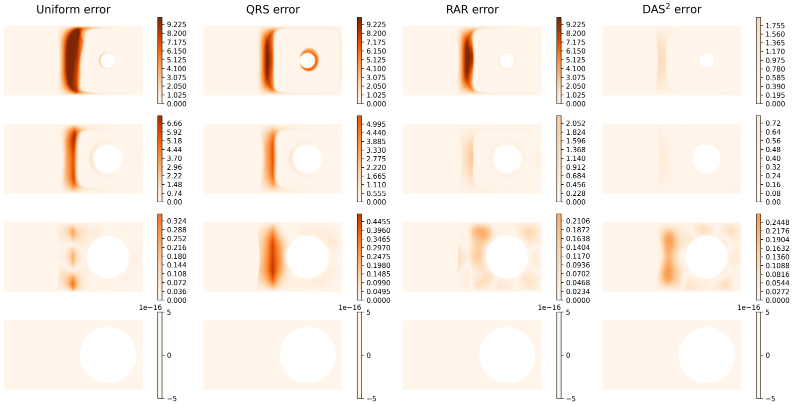
$$l(\mathbf{x},\xi) = x_1(2-x_1)x_2(1-x_2)(\xi_1^2 - (x_1-1.5)^2 - (x_2-0.5)^2).$$

$$u(\mathbf{x},\xi) \approx u_{\theta_u}(\mathbf{x},\xi),\ y(\mathbf{x},\xi) \approx l(\mathbf{x},\xi) y_{\theta_y}(\mathbf{x},\xi) + 1,\ p(\mathbf{x},\xi) \approx l(\mathbf{x},\xi) p_{\theta_p}(\mathbf{x},\xi)$$

$$\Omega := \{(\mathbf{x},\xi) | 0 \le x_1 \le 2, 0 \le x_2 \le 1, 0.05 \le \xi_1 \le 0.45, 0.5 \le \xi_2 \le 2.5,$$
$$(x_1-1.5)^2 + (x_2-0.5)^2 \ge \xi_1^2\}.$$

joint PDF model for sampling

Numerical results: parametric optimal control problems



top to bottom: $\xi = (0.10, 2.5)$ $\xi = (0.20, 2.0)$ $\xi = (0.30, 1.5)$ $\xi = (0.40, 0.5)$.

Numerical results: parametric optimal control problems

sampling strategy \ $\lvert S_\Omega \rvert$	0.5×10^4	1×10^4	1.5×10^4	2×10^4
Uniform (0.1s)	0.92	0.67	0.49	0.29
QRS (0.1s)	0.66	0.63	0.36	0.20
RAR (0.1s)	0.95	0.77	0.37	0.15
DAS^2 (0.1s)	0.89	0.37	0.20	0.06

- 11×11 grid in the parametric space
- dolfin-adjoint solver for a fixed parameter
- dolfin-adjoint solver : 18804 seconds

Parametric lid-driven cavity flow problems

$$\begin{cases} \mathbf{u}(\mathbf{x},\xi)\cdot\nabla\mathbf{u}(\mathbf{x},\xi)+\nabla p(\mathbf{x},\xi)=\dfrac{1}{Re(\xi)}\Delta\mathbf{u}(\mathbf{x},\xi) & \text{in } \Omega, \\ \nabla\cdot\mathbf{u}(\mathbf{x},\xi)=0 & \text{in } \Omega, \\ \mathbf{u}(\mathbf{x},\xi)=\mathbf{g}(\mathbf{x},\xi) & \text{on } \partial\Omega, \end{cases}$$

- $\mathbf{u}(\mathbf{x},\xi)=[u(\mathbf{x},\xi),v(\mathbf{x},\xi)]^{\mathsf{T}}, \mathbf{x}=[x,y]^{\mathsf{T}}$
- $Re(\xi)=\xi\in\Omega_p=[100,1000]$
- The physical domain is $\Omega_s=[0,1]\times[0,1]$
- Boundary conditions

$$\mathbf{g}(\mathbf{x},\xi)=\begin{cases}[1,0]^{\mathsf{T}}, y=1; \\ [0,0]^{\mathsf{T}}, \text{ otherwise.}\end{cases}$$

Goal: obtaining all-at-once solutions for $Re\in[100,1000]$

Parametric lid-driven cavity flow problems

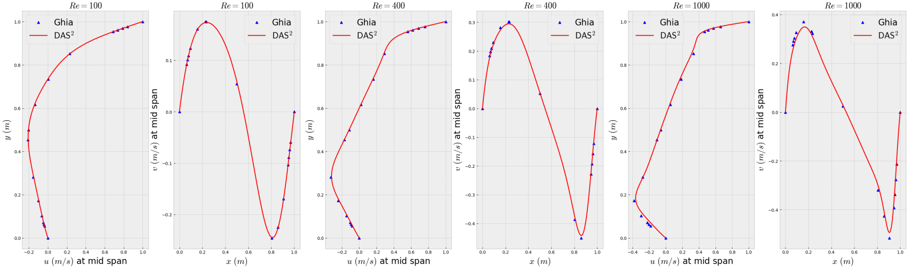
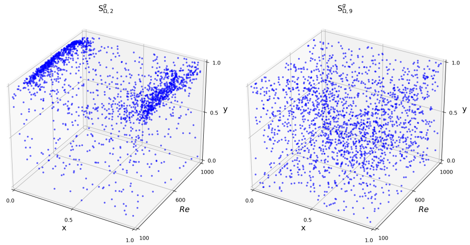
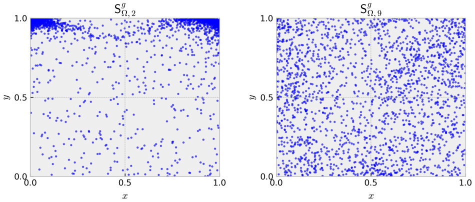


Figure: The velocity components at the location of mid-span lines for surrogate modeling of parametric lid-driven cavity flow problems ($Re \in [100, 1000]$). The results for $Re = 100, 400, 1000$ are chosen for visualization.

Parametric lid-driven cavity flow problems



(a) 3d points



(b) 2d projection onto *xy*-plane

Parametric lid-driven cavity flow problems

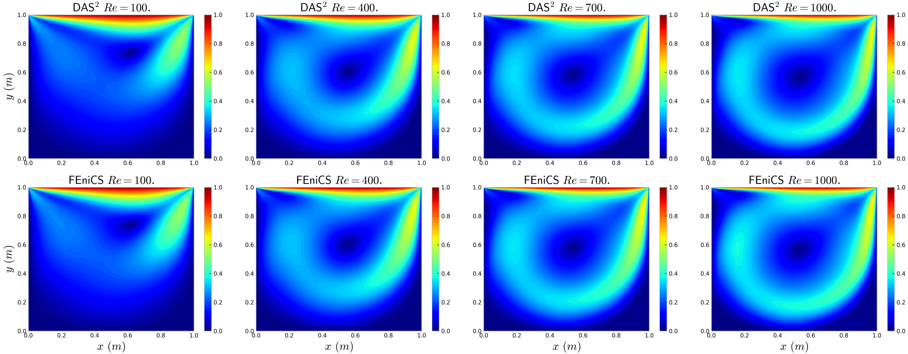


Figure: The visualization of $|\mathbf{u}| = \sqrt{u^2 + v^2}$ for surrogate modeling of parametric lid-driven cavity flow problems, $Re \in [100, 1000]$. The l_2 relative errors are $1.5\%, 1.1\%, 3.1\%, 4.8\%$ for $Re = 100, 400, 700, 1000$ respectively.

- Inference time of DAS: 0.02 seconds,
- The computation time of FEniCS: 309.94 seconds.

Summary and outlook

summary

- illustrate that DAS is necessary for constructing surrogate models
- significantly improve the accuracy for PDEs with low-regularity problems especially for high-dimensional or parametric problems
- DAS, a general and flexible framework for the adaptive learning strategy

outlook

- incoporate tensor networks into deep adaptive sampling
- large scale problems
- more applications

Thank you for your attention

Q & A