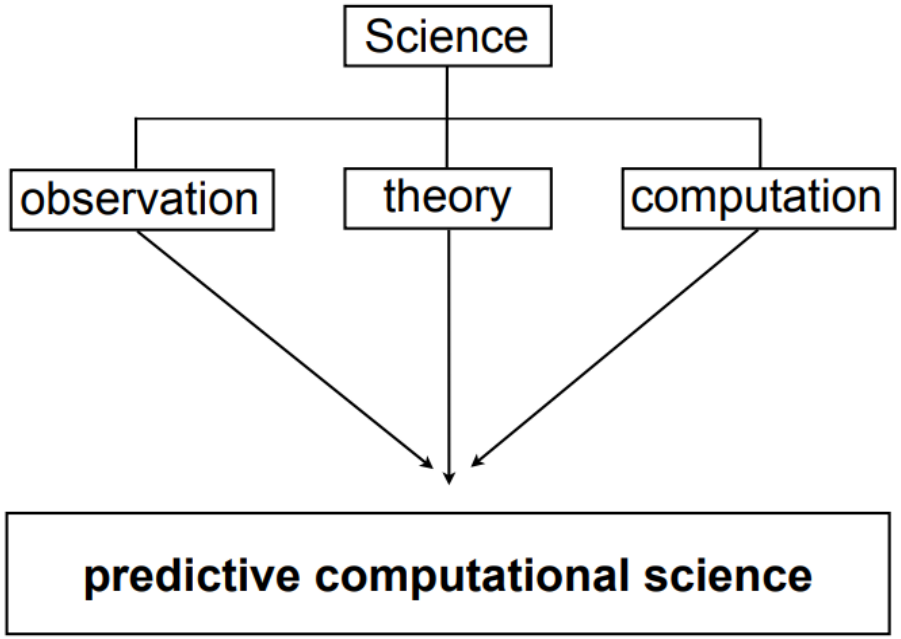
Deep adaptive sampling: Algorithm, Theory, and Applications

Peng Cheng Laboratory

Kejun Tang
tangkj@pcl.ac.cn
August 9, 2022
Joint work with Xiaoliang Wan (Louisiana State University) and Chao Yang (Peking University)

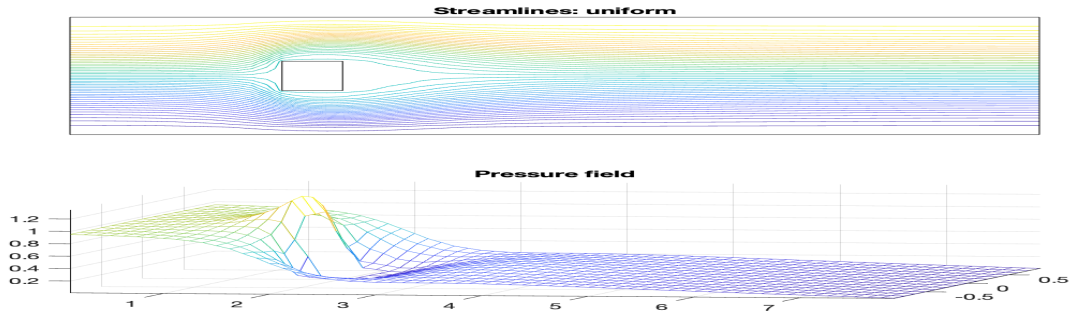
Background



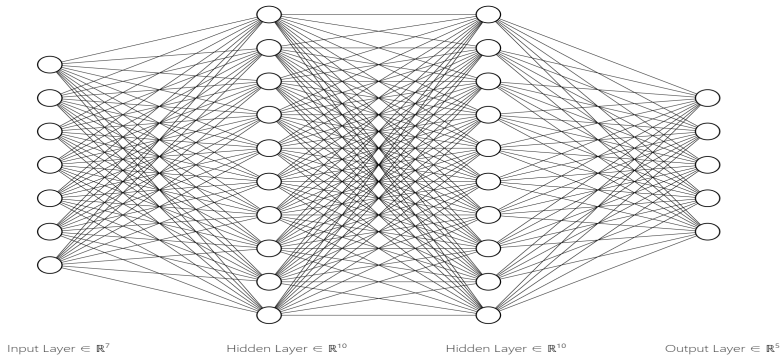
- Diffusion process
- Aerospace
- Molecular dynamics
- ...

Background

- Mathematical (physical) model: PDEs or ODEs

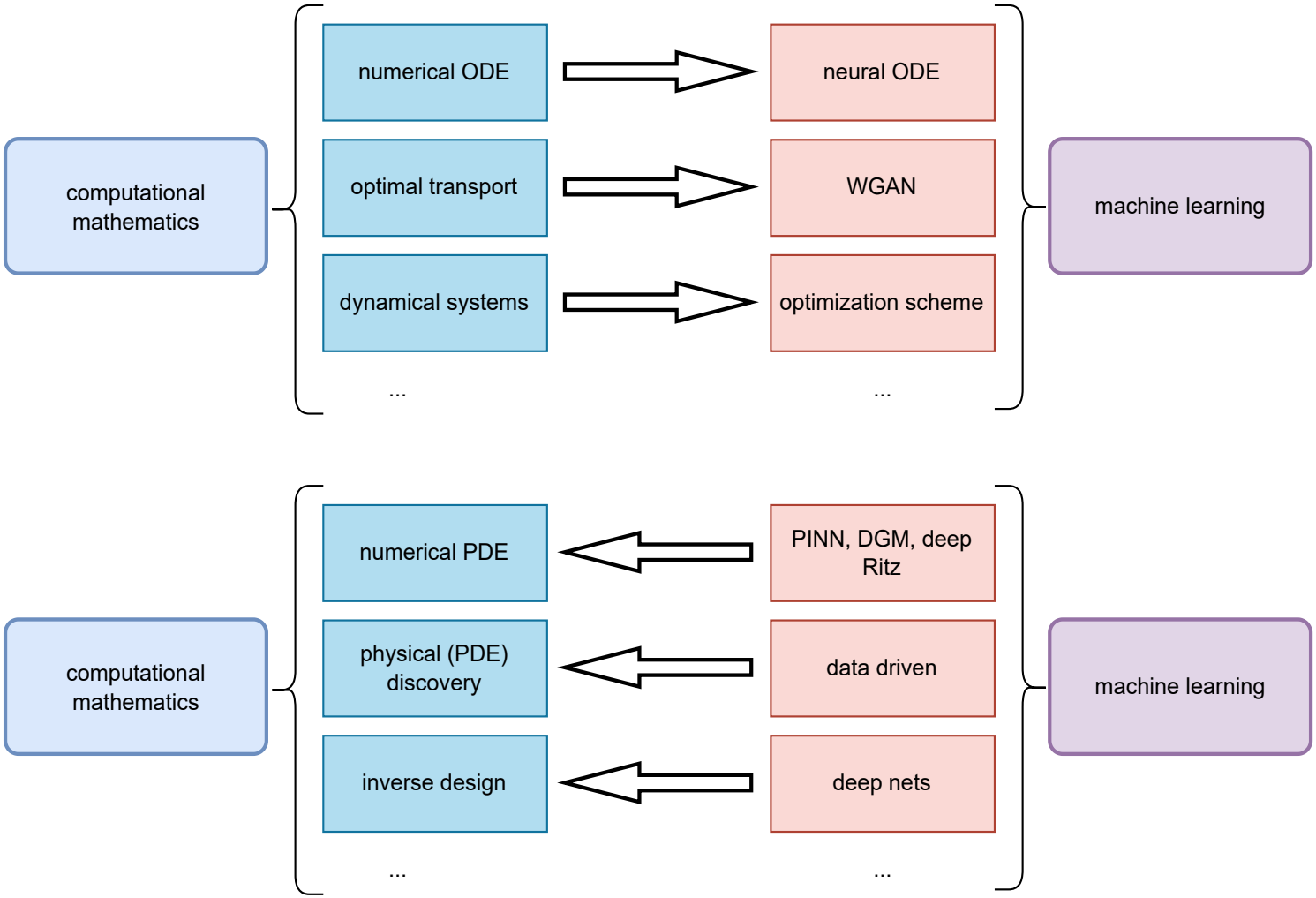


- Data-driven model (e.g., deep neural networks): no proper physical model but massive available data



- Numerical methods
  Both 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]
- ...

Big data era: data-driven

Using data to train a predictive model with parameters Θ

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

e.g. deep neural networks

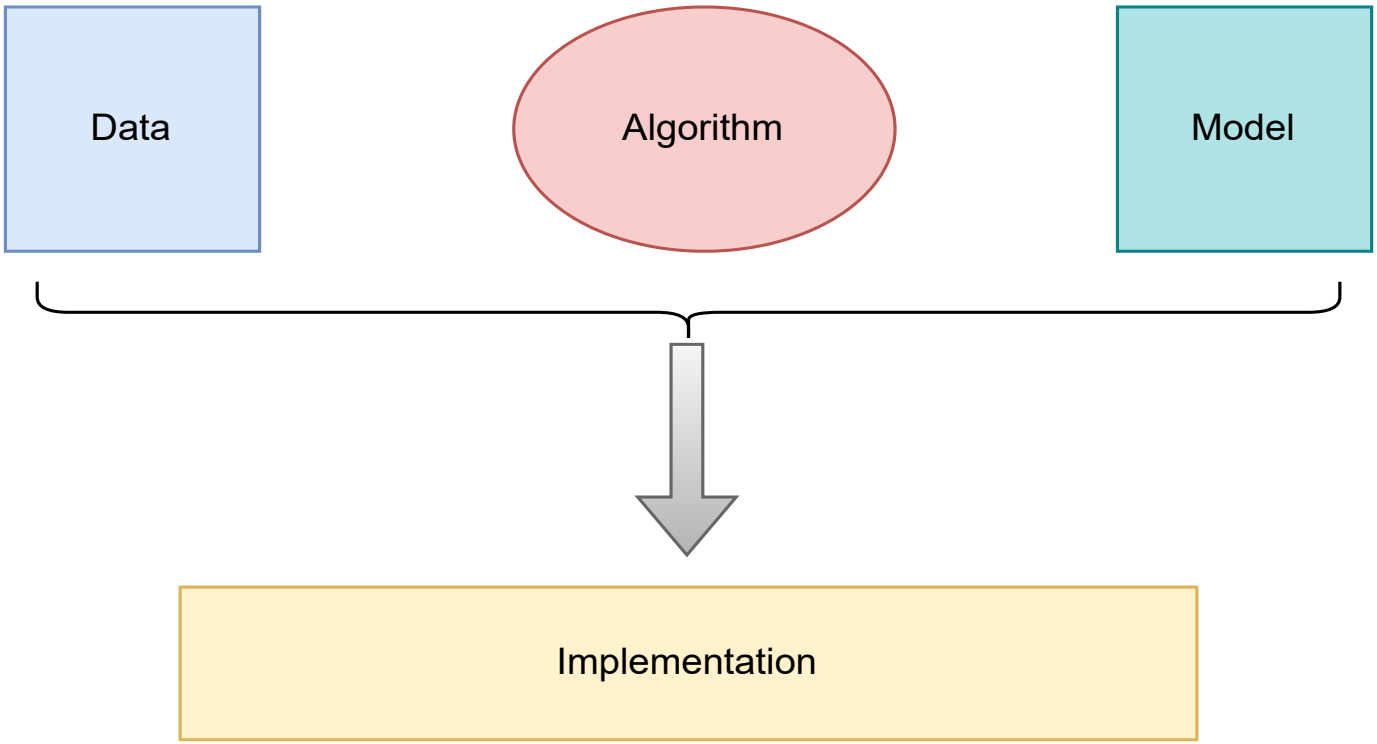
Training usually means an optimization problem

$$\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} J(x^{(i)}; \Theta).$$

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

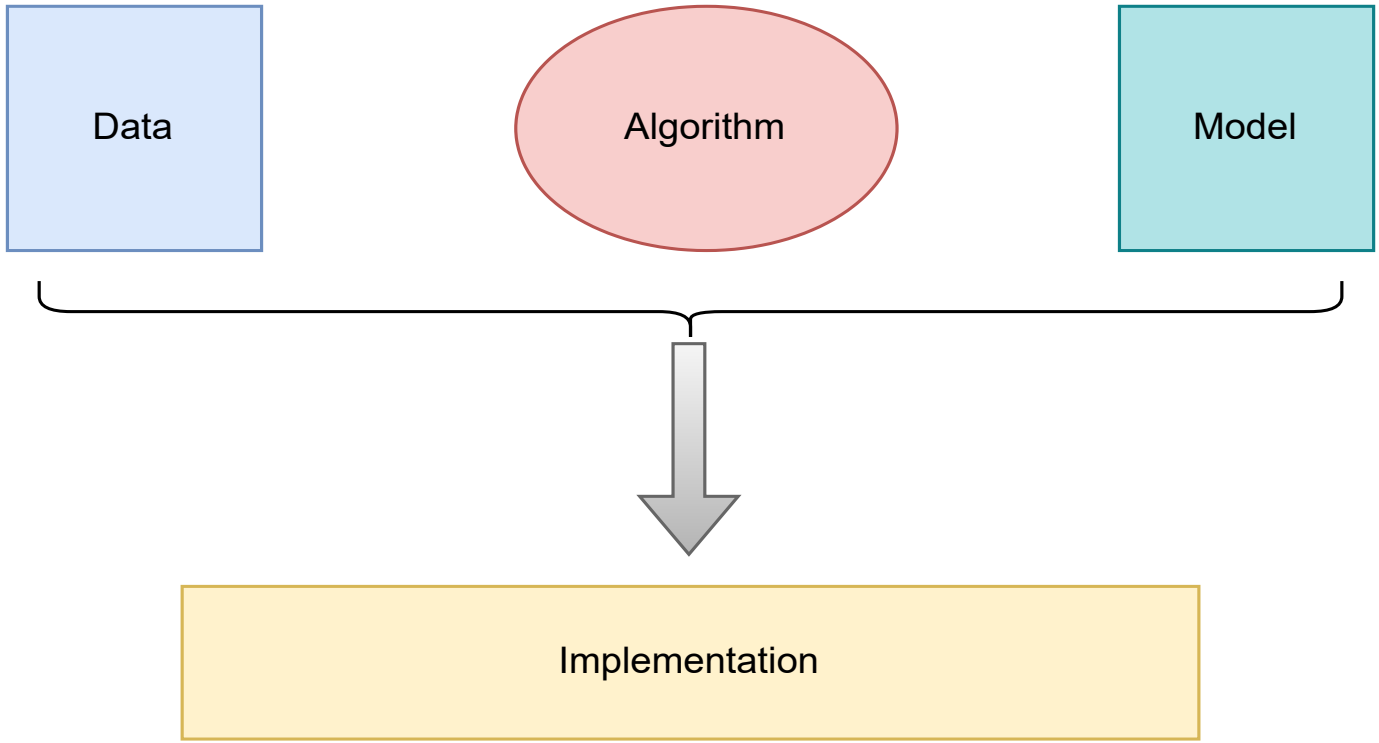
- machine learning
- computer vision
- signal processing
- ...

Big data era: data-driven



- Model: deep neural networks, physical models, or coupling
- Data: labeled, unlabled, random samples (our case) ....
- 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 scientific computing problems (e.g., Fokker-Planck equations , committor functions approximation)

- deep networks to alleviate the curse of dimensionality
- develop adaptive schemes using machine learning techniques

Illustration of the statistical error

illustrate the statistical error of the machine learning technique from a function approximation perspective

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

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

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

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

Illustration of the statistical error

illustrate the statistical error of the machine learning technique from a function approximation perspective

For a linear space $V = \mathrm{span}\{q_i(\mathbf{x})\}_{i=1}^n$

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

Lemma (Tang, Wan and Yang, 2022)

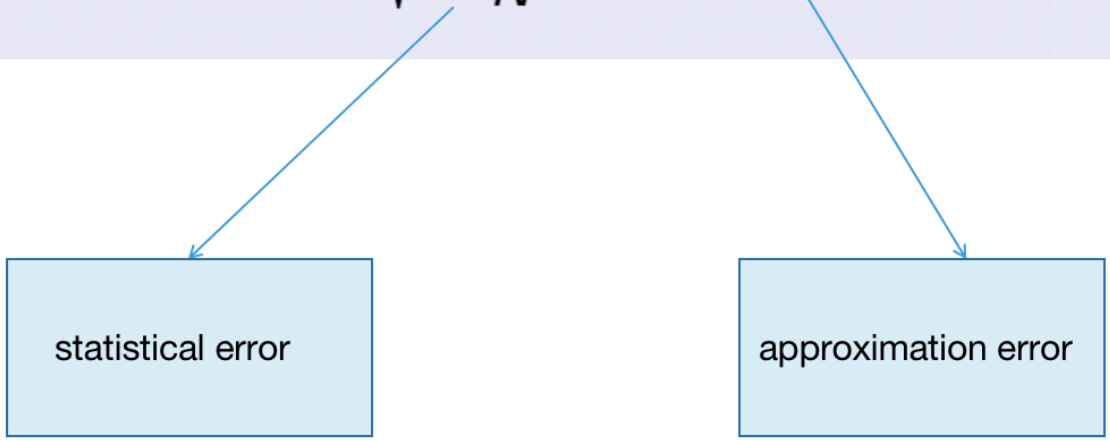
Let $h(\mathbf{x}) \in C(D)$ be a continuous function defined on a compact domain $D \subset \mathbb{R}^d$ and $\rho(\mathbf{x}) > 0$ be a PDF on D. Let $V = \mathrm{span}\{q_i(\mathbf{x})\}_{i=1}^n$ with $q_i(\mathbf{x})$ being orthonormal polynomials in terms of $\rho(\mathbf{x})$. For any $\delta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

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

where C is a constant, and $\|\cdot\|_\rho$ is the weighted L_2 norm in terms of $\rho(\mathbf{x})$.

Illustration of the statistical error

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



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

Partial differential equations

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

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

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

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

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

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

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

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

Key point: $\min_\Theta J(u(\mathbf{x}; \Theta)) \to \min_\Theta J_N(u(\mathbf{x}; \Theta))$ 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

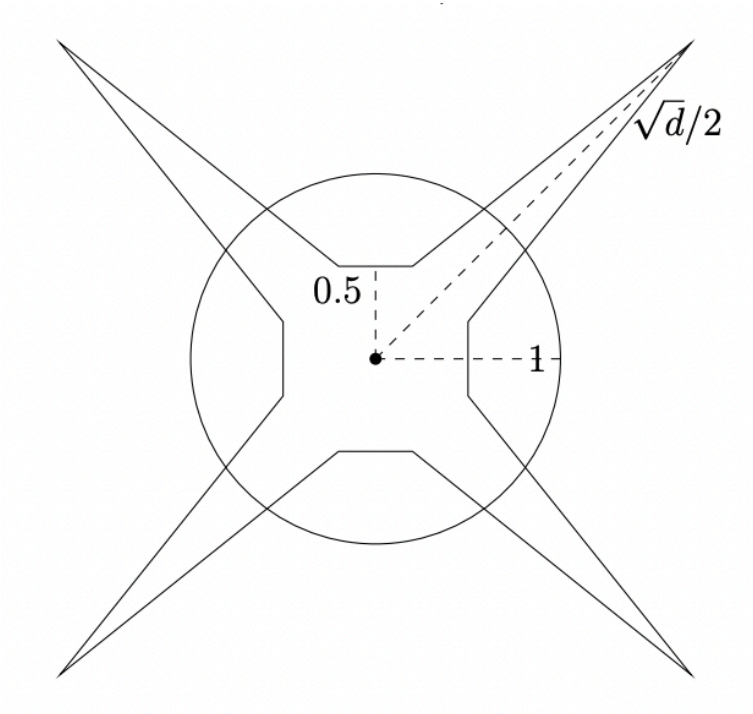
the capability of neural networks $\rightarrow$ approximation error

the strategy of loss discretization $\rightarrow$ statistical error

Key point: how to sample?

Geometric properties of high-dimensional spaces

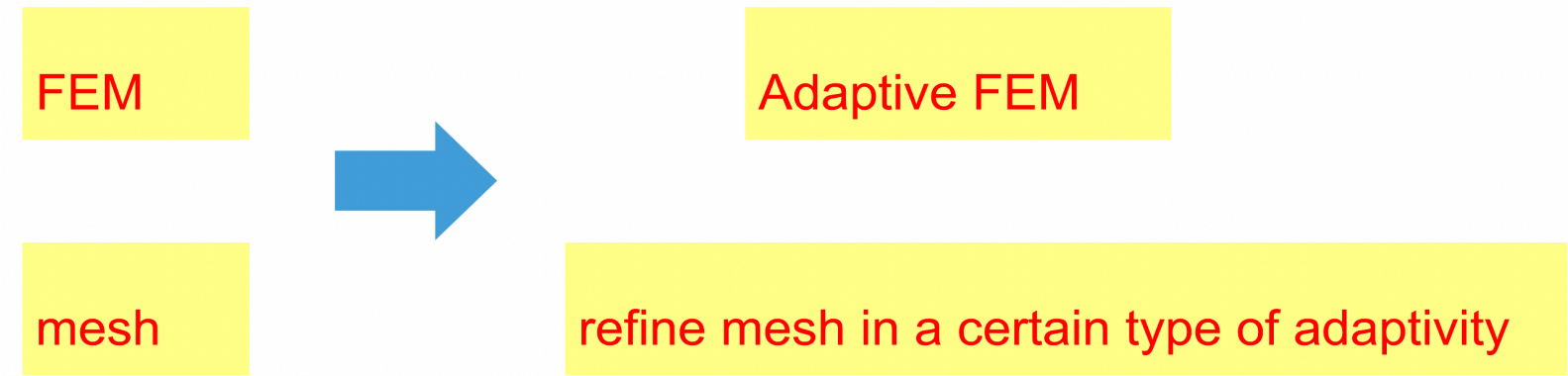
uniformly distributed points in high-dimensional spaces



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 $O(1/\zeta) \rightarrow$ 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*

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

where

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

and

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

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]

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

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

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

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

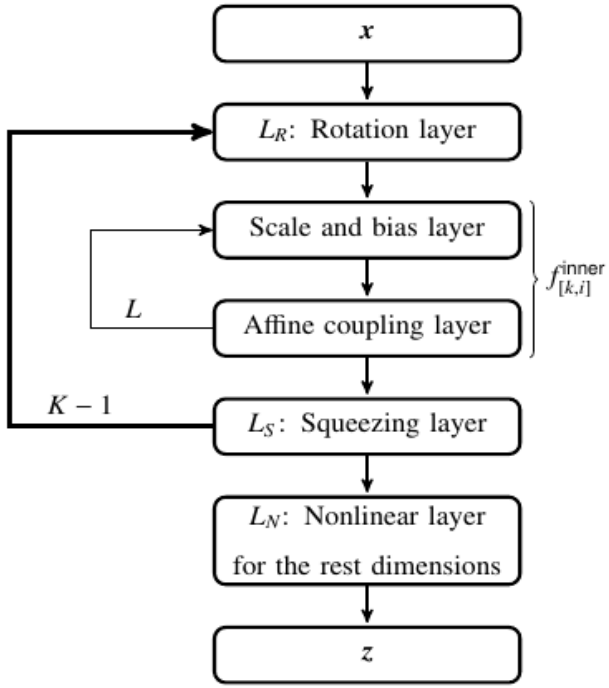
Advantages

- 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

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, 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)

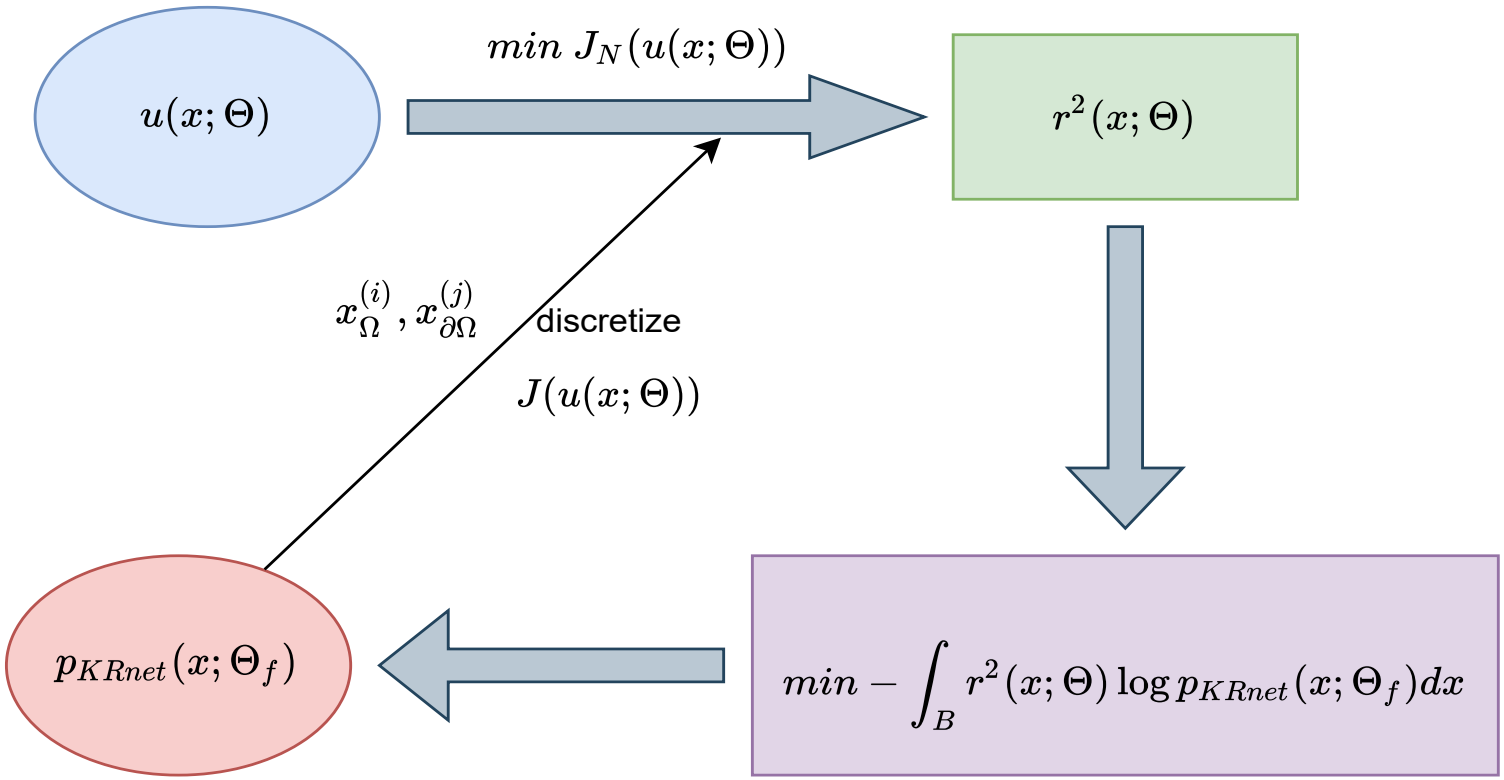
Generate $\mathbf{x}_{\Omega,k+1}^{(i)} \subset \Omega$ through $p_{KRnet}(\mathbf{x};\Theta_f^{*,(k+1)})$.

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).

Deep adaptive sampling method (DAS)

The framework of DAS. (see [Tang, Wan and Yang, 2022] for more details) [2]



[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 for PDEs

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*

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

Some progresses

Fokker-Planck equations

- Develop adaptive deep learning techniques for solving Fokker-Planck equations

Adaptive deep density approximation for Fokker-Planck equations, Journal of Computational Physics, 2022.

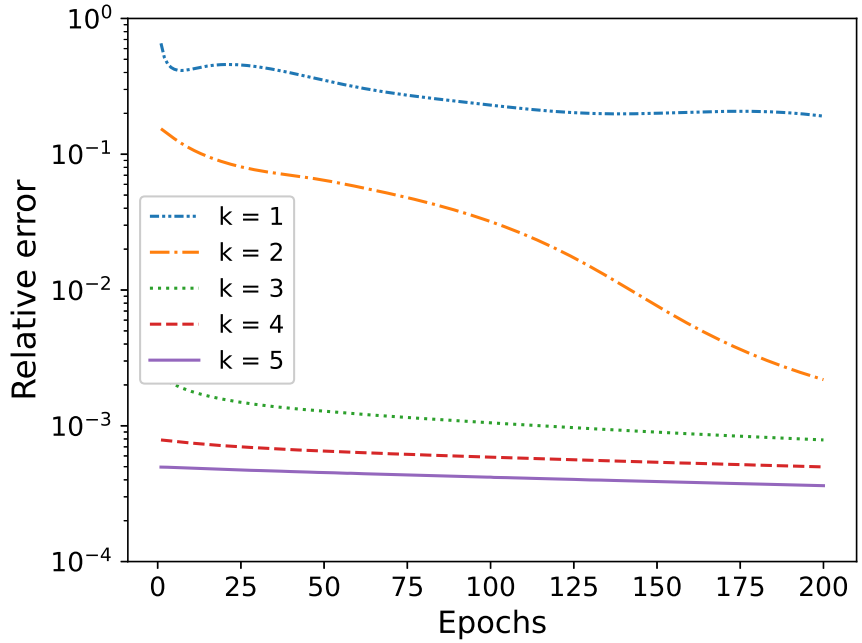
General high-dimensional PDEs

- Develop adaptive deep learning techniques for solving general PDEs

DAS: A deep adaptive sampling method for solving partial differential equations, preprint, 2022.

Fokker-Planck equations

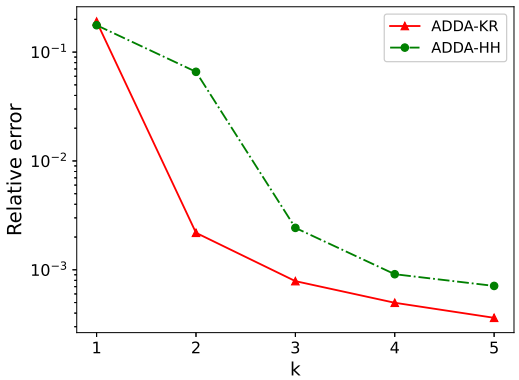
A special case: $u(\mathbf{x}) = p(\mathbf{x})$



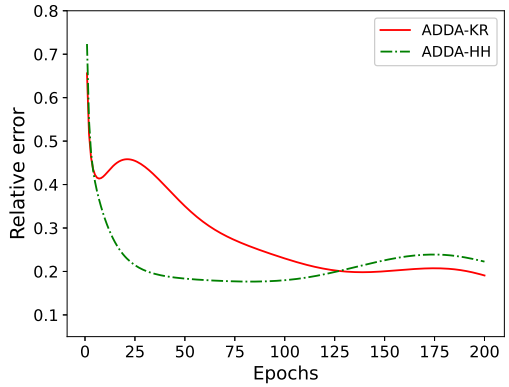
setting

- $\frac{\partial p(\mathbf{x},t)}{\partial t} = \nabla \cdot [p(\mathbf{x},t)\nabla \log(\beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}))] + \nabla^2 p(\mathbf{x},t)$
- 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

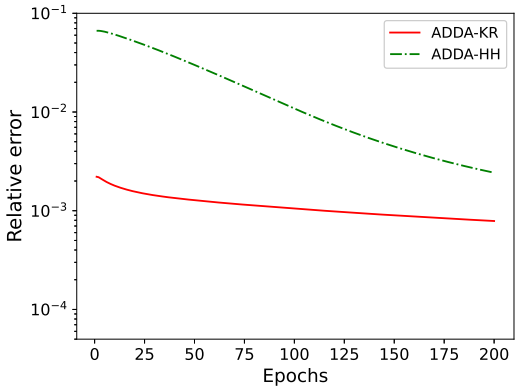
Fokker-Planck equations



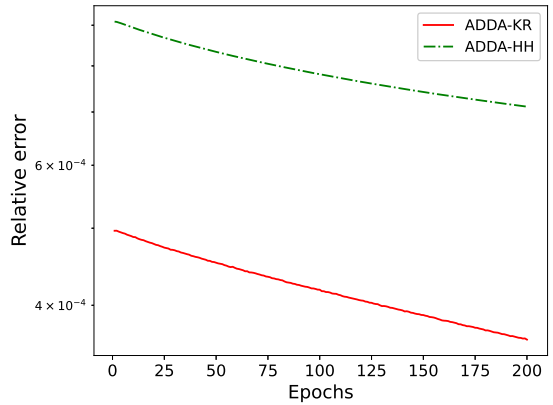
(a) KL divergece w.r.t. k-th model.



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

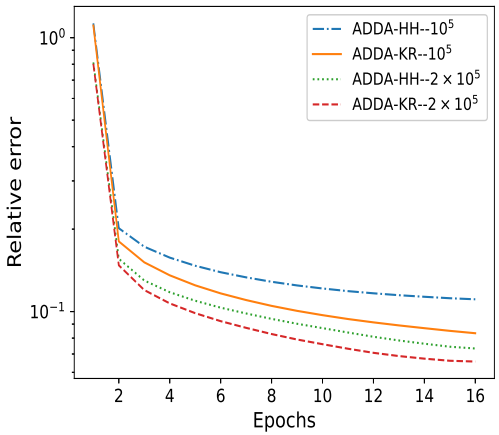


(c) The convergence behavior for $k = 3$.

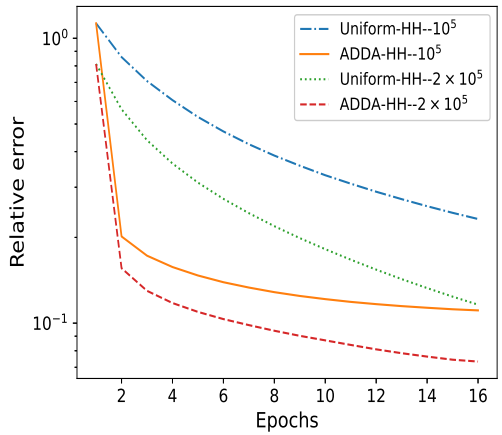


(d) The convergence behavior for $k = 5$.

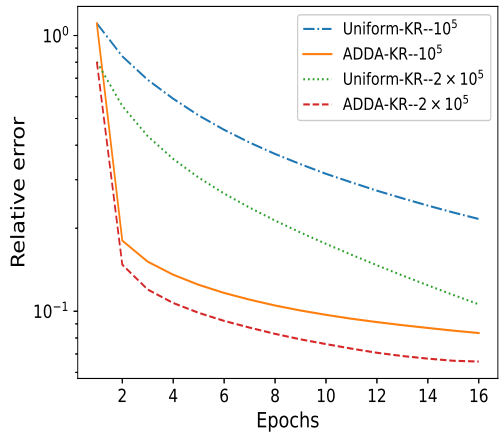
Fokker-Planck equations



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



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

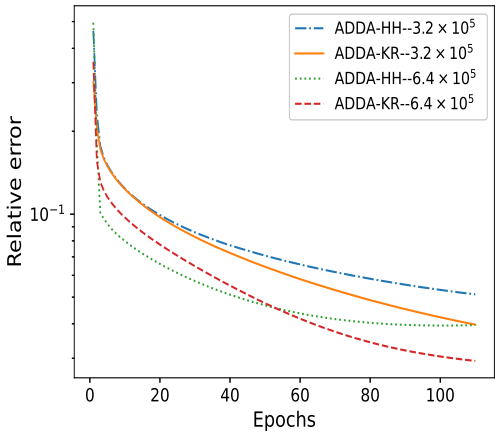


(g) KL-divergence w.r.t epochs for KR

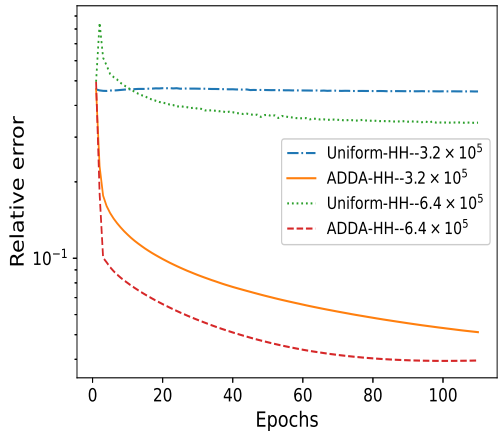
setting (HH: Real NVP)

- $\frac{\partial p(\mathbf{x},t)}{\partial t} = \nabla \cdot [p(\mathbf{x},t)\nabla \log(\beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}))] + \nabla^2 p(\mathbf{x},t)$
- 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

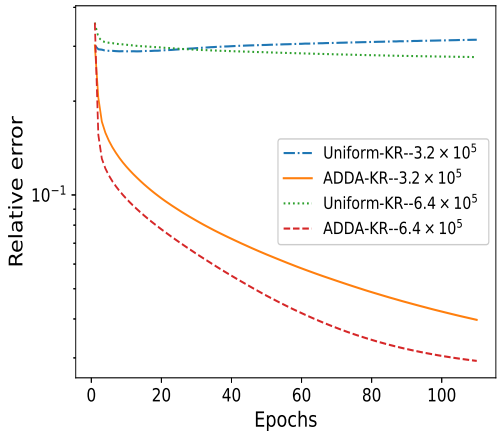
Fokker-Planck equations



(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

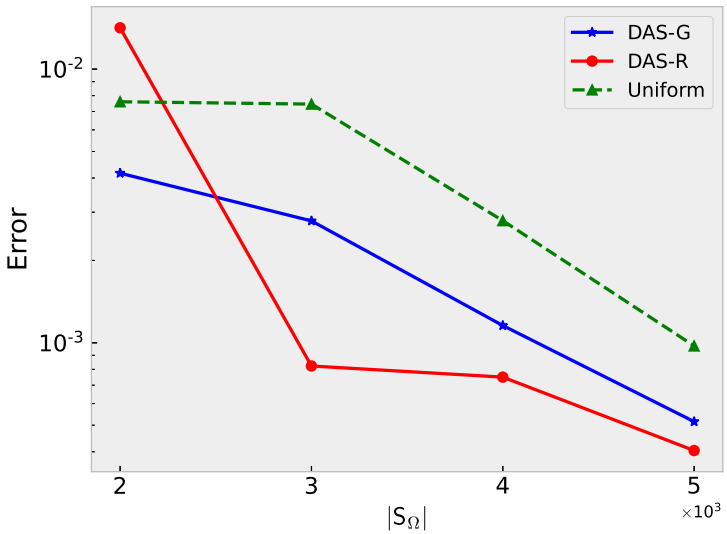
setting

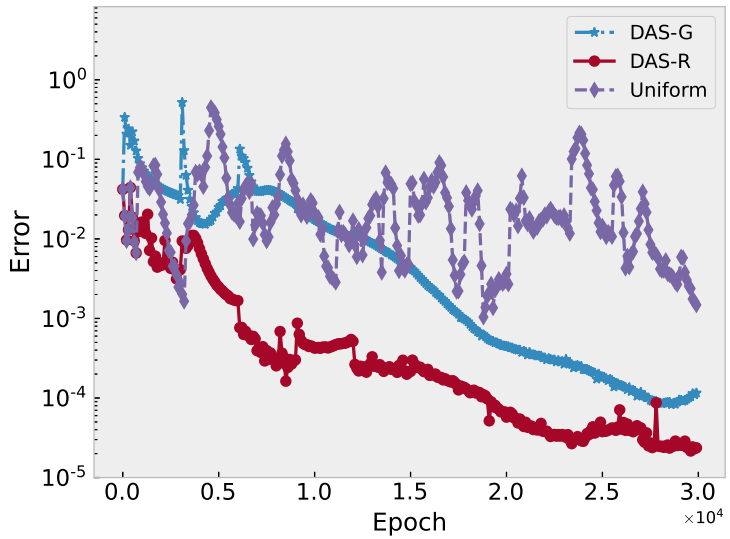
- $\frac{\partial p(\mathbf{x},t)}{\partial t} = \nabla \cdot [p(\mathbf{x},t)\nabla \log(\beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}))] + \nabla^2 p(\mathbf{x},t)$
- stationary solution
  $p_{st}(\mathbf{x}) = \beta_1 p_1(\mathbf{x}) + \beta_2 p_2(\mathbf{x}), \mathbf{x} \in \mathbb{R}^8, p_i(\mathbf{x})$: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

Elliptic PDEs: low-dimensional and low regularity cases

Two-dimensional peak problem

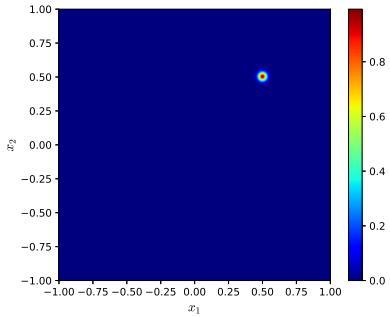
$$
\begin{aligned}
-\Delta u(x_1, x_2) &= s(x_1, x_2) \quad \text{in } \Omega, \\
u(x_1, x_2) &= g(x_1, x_2) \quad \text{on } \partial\Omega,
\end{aligned}
$$



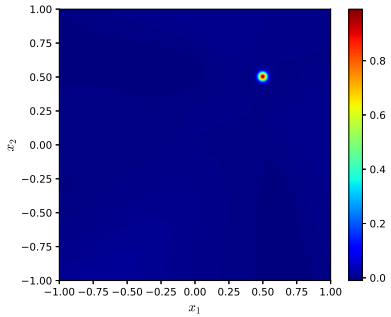


Elliptic PDEs: low-dimensional and low regularity cases

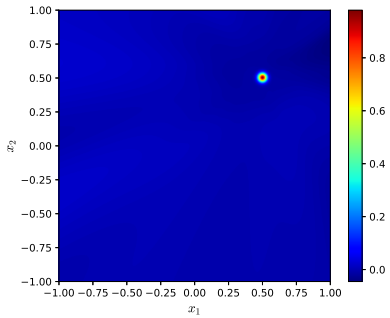
Two-dimensional peak problem



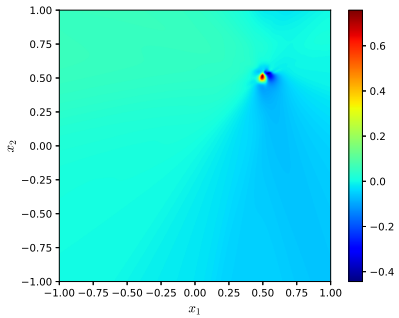
(k) The exact solution.



(l) DAS-R approximation.



(m) DAS-G approximation.

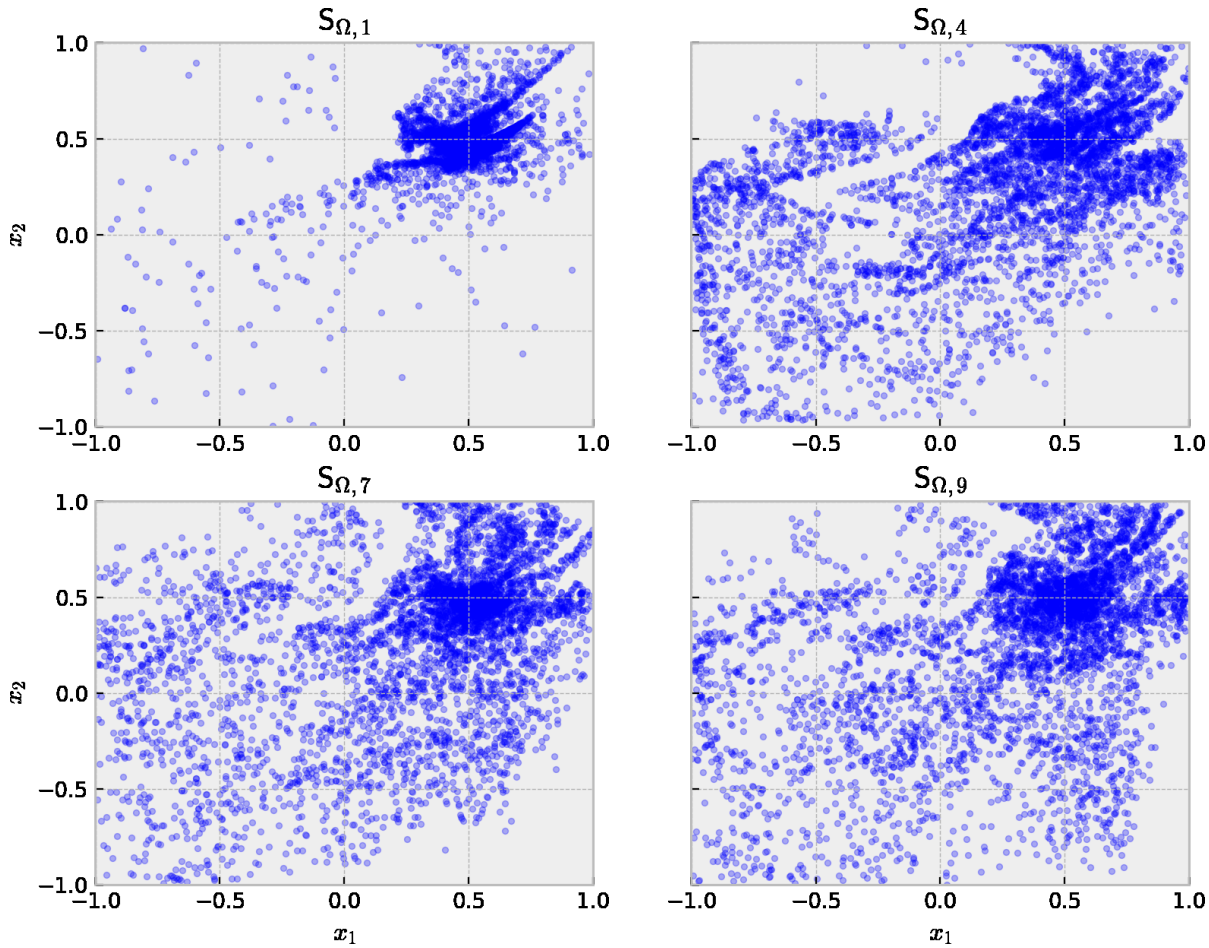


(n) Uniform sampling strategy.

Elliptic PDEs: low-dimensional and low regularity cases

Two-dimensional peak problem

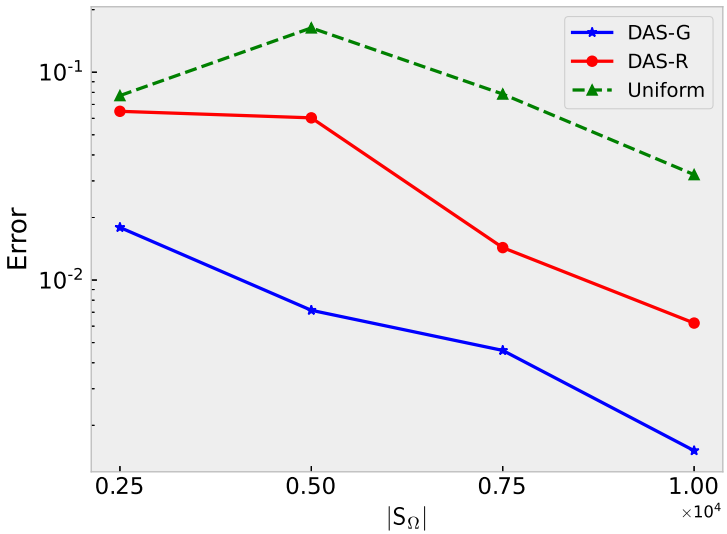
DAS-R samples

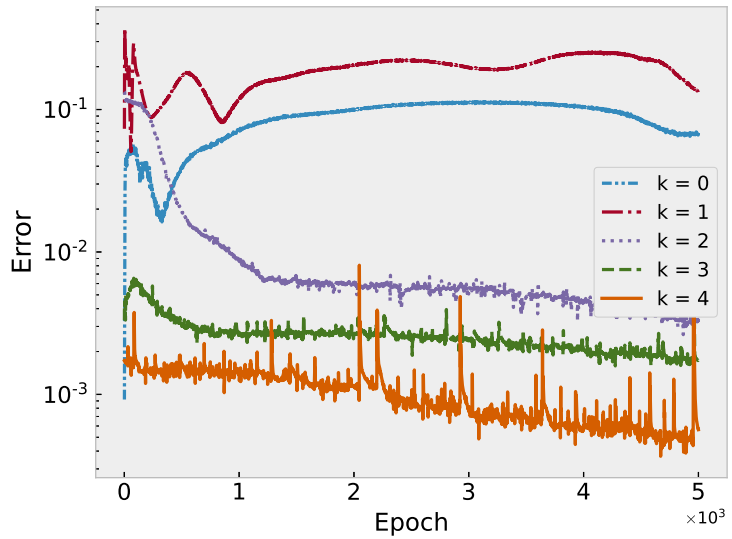


Elliptic PDEs: low-dimensional and low regularity cases

Two-dimensional problem with two peaks

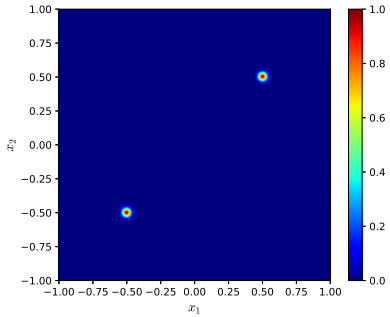
$$-\nabla \cdot \left[u(x_1, x_2)\nabla(x_1^2 + x_2^2)\right] + \nabla^2 u(x_1, x_2) = s(x_1, x_2) \quad \text{in } \Omega,$$
$$u(x_1, x_2) = g(x_1, x_2) \quad \text{on } \partial\Omega,$$



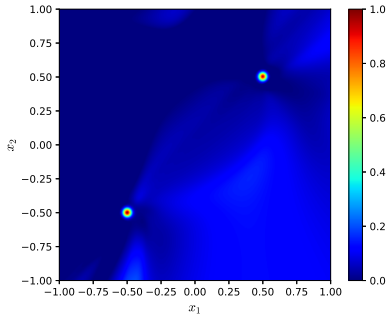


Elliptic PDEs: low-dimensional and low regularity cases

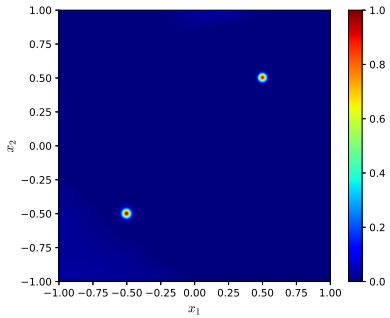
Two-dimensional problem with two peaks



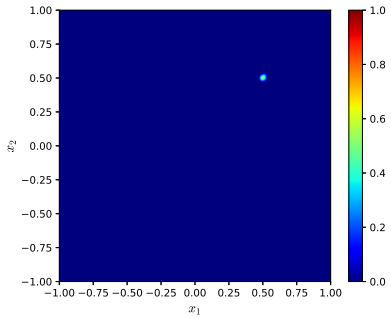
(o) The exact solution.



(p) DAS-R approximation.



(q) DAS-G approximation.

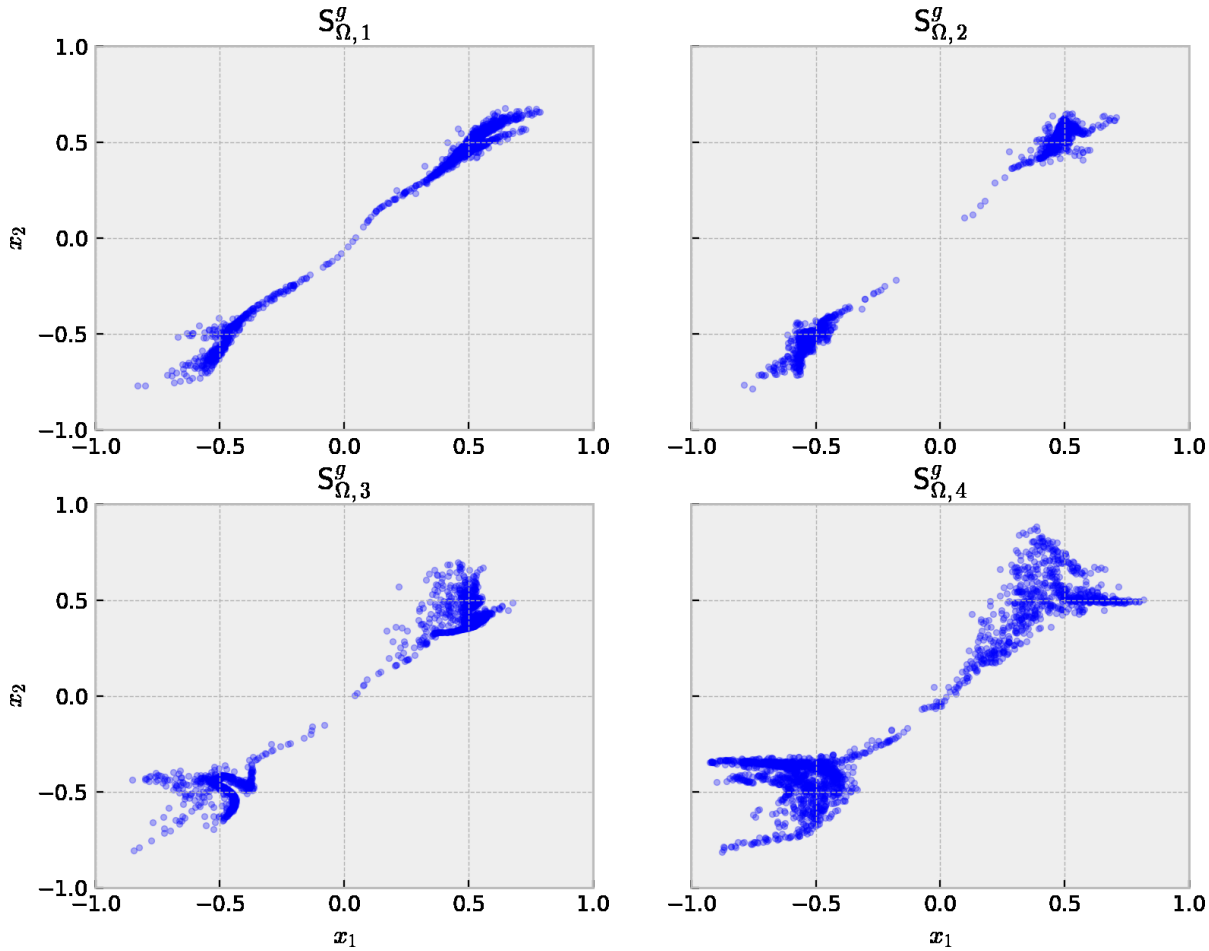


(r) Uniform sampling strategy.

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

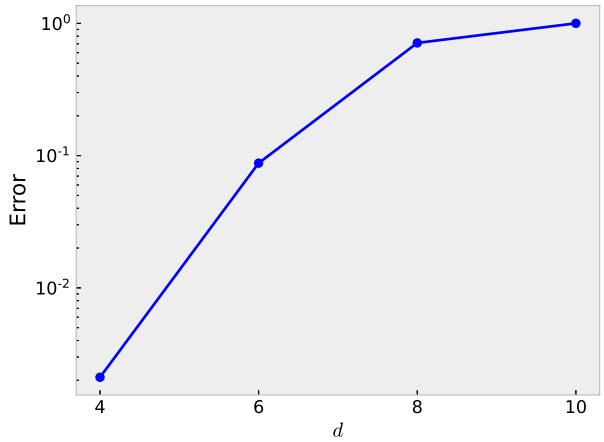
The d-dimensional linear equation

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

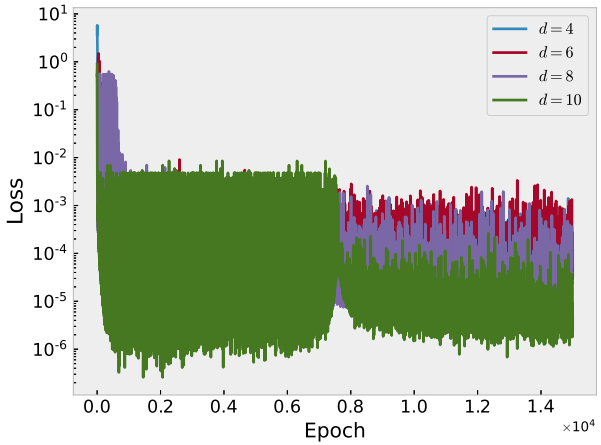
with an exact solution

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

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



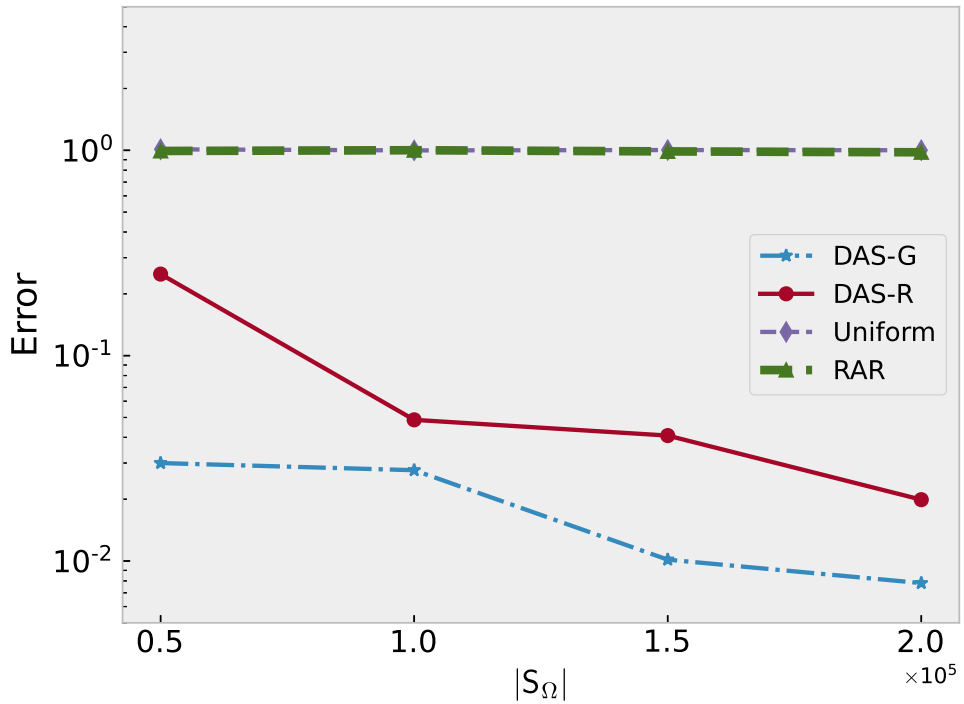
(s) Error



(t) Loss

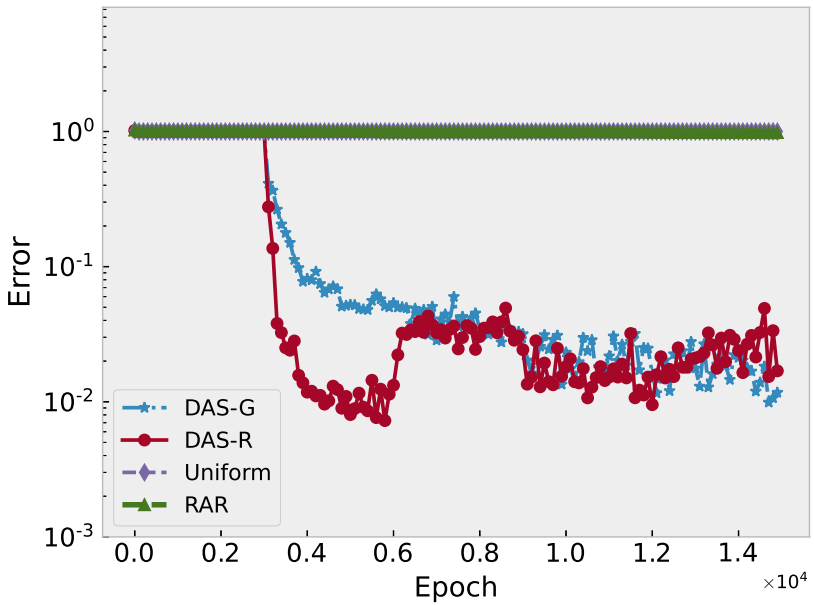
Linear PDEs: High-dimensional and low regularity cases

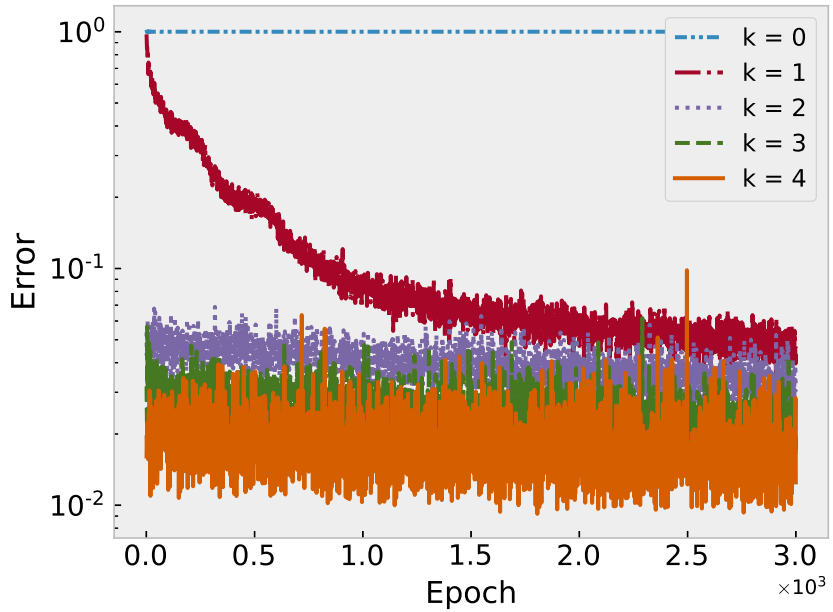
The 10-dimensional linear equation



Linear PDEs: High-dimensional and low regularity cases

The 10-dimensional linear equation

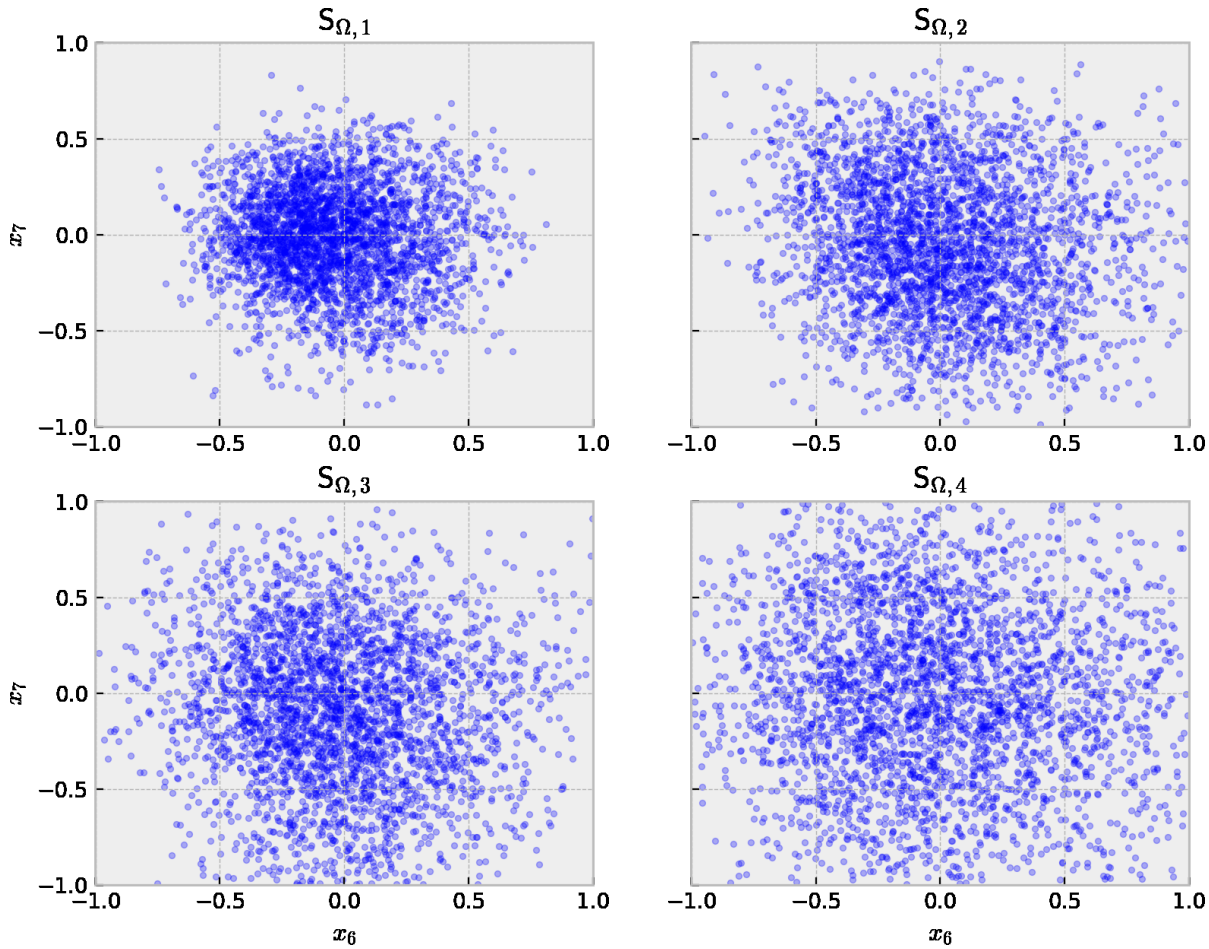




Linear PDEs: High-dimensional and low regularity cases

The 10-dimensional linear equation

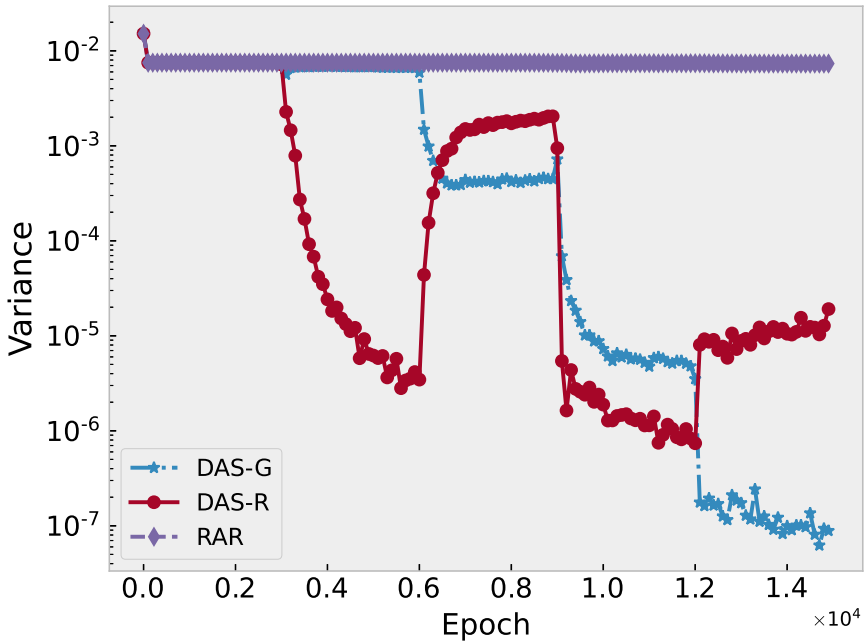
DAS-R samples



Linear PDEs: High-dimensional and low regularity cases

The 10-dimensional linear equation

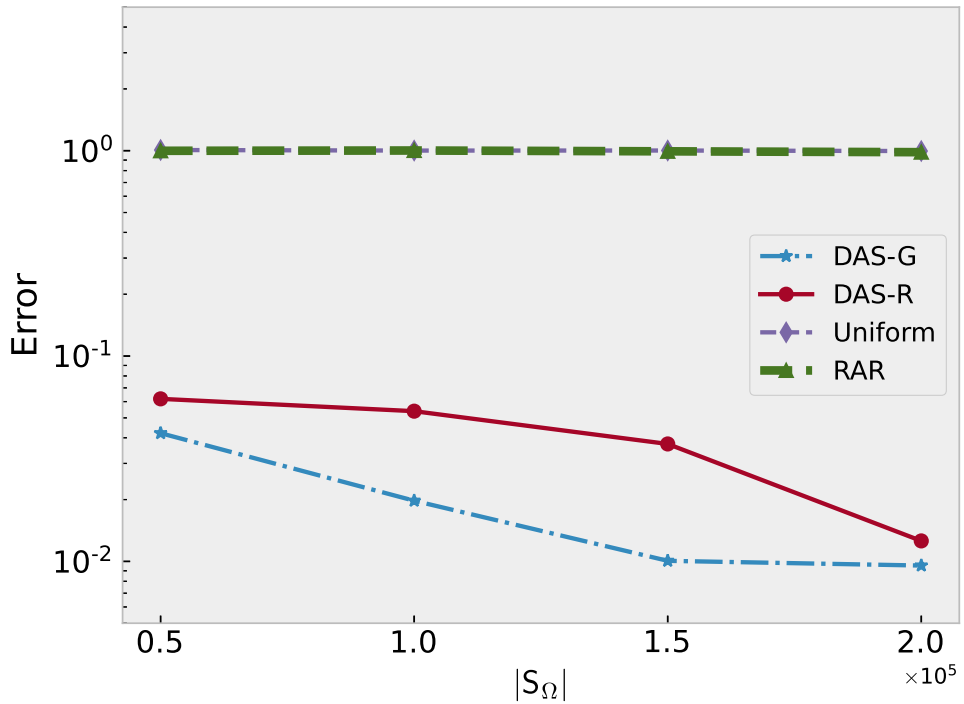
The evolution for the variance of residual



Nonlinear PDEs: High-dimensional and low regularity cases

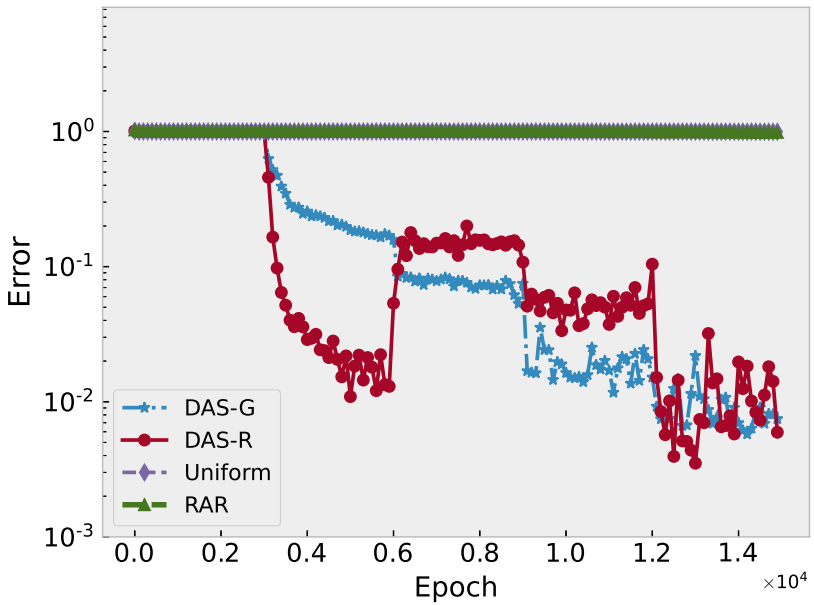
The 10-dimensional nonlinear equation

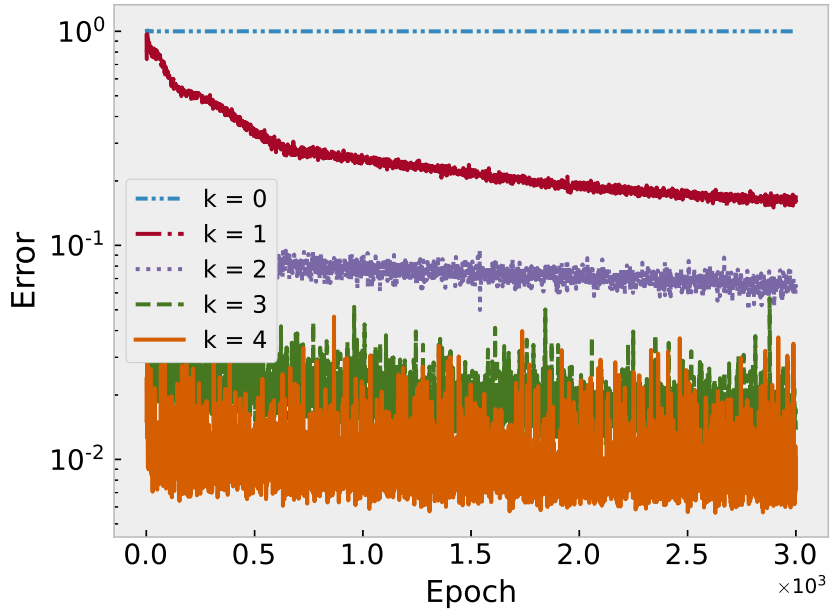
$$-\Delta u(\mathbf{x}) + u(\mathbf{x}) - u^3(\mathbf{x}) = s(\mathbf{x}), \quad \mathbf{x} \text{ in } \Omega = [-1, 1]^{10}.$$



Nonlinear PDEs: High-dimensional and low regularity cases

The 10-dimensional nonlinear equation

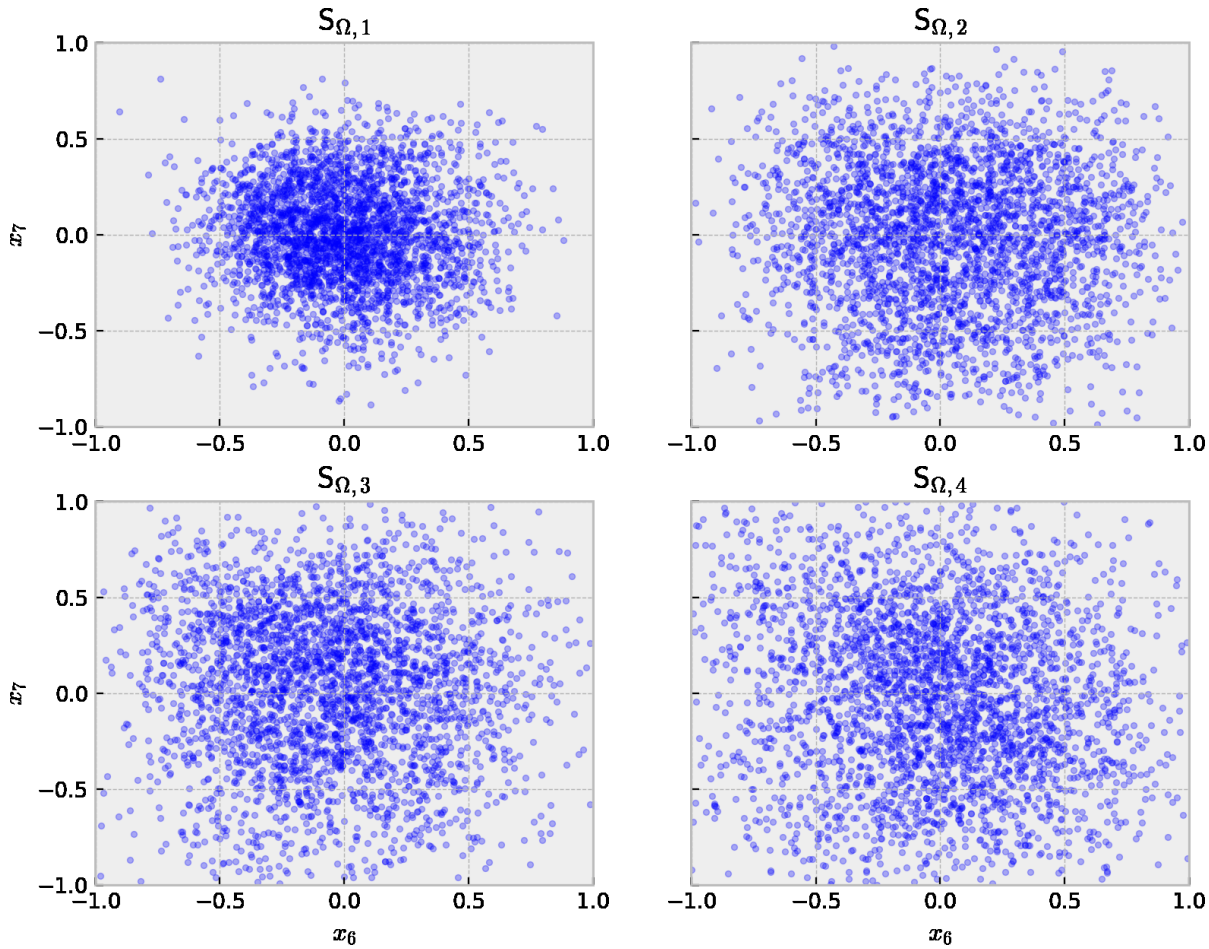




Nonlinear PDEs: High-dimensional and low regularity cases

The 10-dimensional nonlinear equation

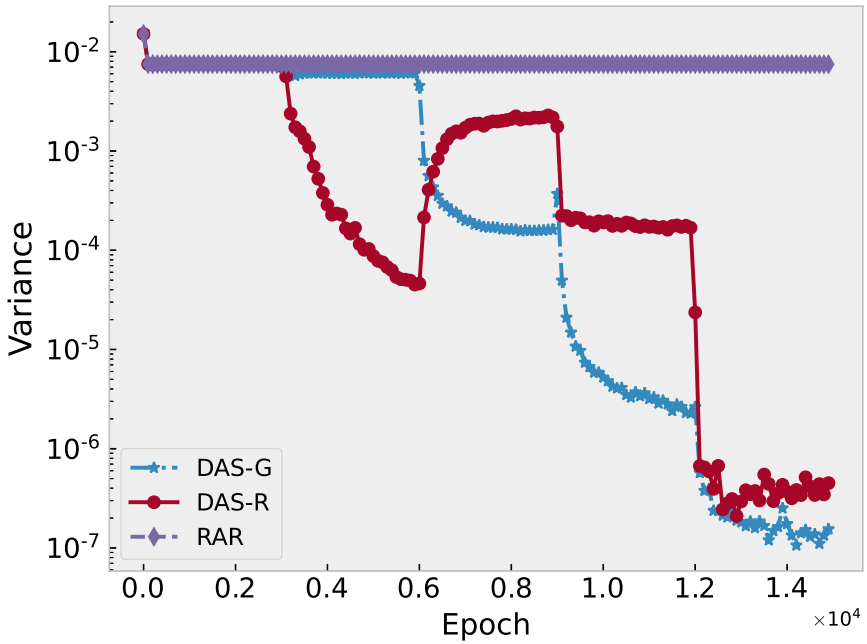
DAS-R samples



Elliptic PDEs: High-dimensional and low regularity cases

The 10-dimensional nonlinear equation

The evolution for the variance of residual



Summary and outlook

summary

- a general and flexible adaptive learning strategy using deep generative models for sample generation
- significantly improve the accuracy for PDEs with low regularity problems especially when the dimensionality is relatively large

outlook

- large scale problems
- more robust and efficient PDF approximation and sample generation
- realistic applications