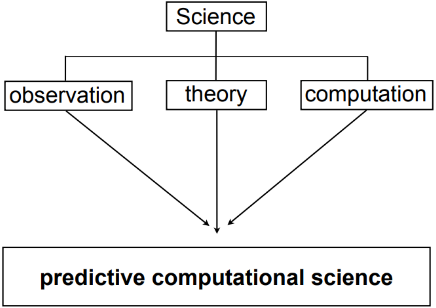
DAS: A deep adaptive sampling methd for solving partial differential equations

Peng Cheng Laboratory

Kejun Tang
tangkj@pcl.ac.cn

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

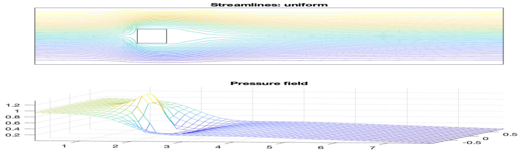
Background



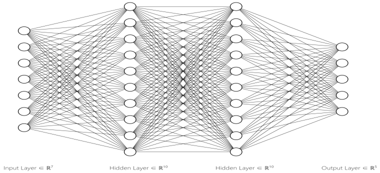
- Electromagnetic field
- Aerospace
- Molecular dynamics
- ...

Background

- Mathematical (physical) model: PDEs or ODEs

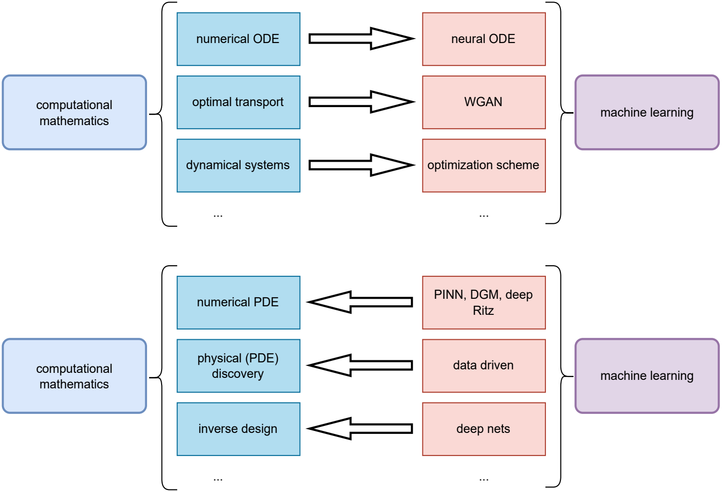


- Data-driven model (e.g., deep neural networks): no proper physical model but massive available data



- 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

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

$\mathcal{L}$: 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

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

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

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

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

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

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

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

Deep learning for PDE

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

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

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

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

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

Key point: $\min\limits_{\Theta} J\left(u(\mathrm{x};\Theta)\right) \to \min\limits_{\Theta} J_N\left(u(\mathrm{x};\Theta)\right)$ discretize loss by uniform sampling

Deep learning for PDE

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

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

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

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?

FEM

mesh

Adaptive FEM

refine mesh in a certain type of adaptivity

Observation: uniform mesh is not optimal for FEM

Deep method

lack of adaptivity $\rightarrow$ 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

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

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

or relax the definition of $J_r(u)$

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

Importance sampling

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

Deep adaptive sampling method (DAS)

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

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

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

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

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

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

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]

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

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

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

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

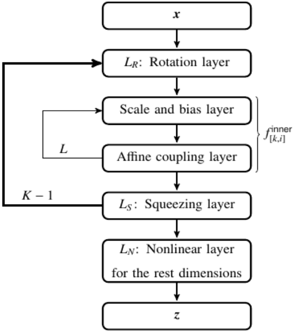
Advantages

- 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)

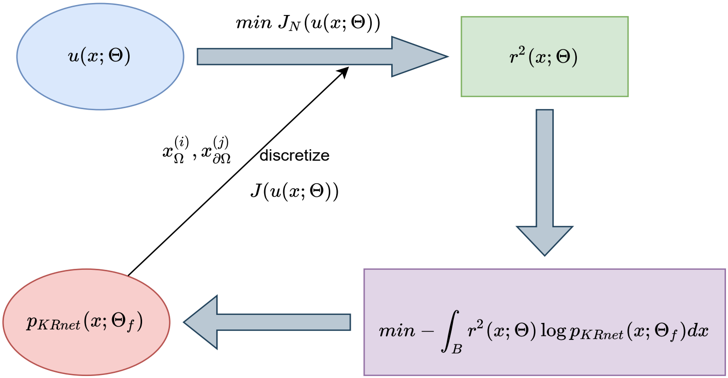
The framework of DAS (see [Tang, Wan and Yang, 2021] for more details) [1]

// solve PDE
Sample m samples $\mathrm{x}_{\Omega,k}^{(i)}$ and Sample m samples $\mathrm{x}_{\partial\Omega,k}^{(j)}$.
Update $u(\mathrm{x};\Theta)$ by descending the stochastic gradient of $J_N(u(\mathrm{x};\Theta))$.
// Train KRnet
Sample m samples from $\mathrm{x}_{\Omega,k}^{(i)}$.
Update $p_{KRnet}(\mathrm{x};\Theta_f)$ by descending the stochastic gradient of $H(\hat{r}_X, \hat{p}_{KRnet})$.
// Refine training set
Generate $\mathrm{x}_{\Omega,k+1}^{(i)} \subset \Omega$ through $p_{KRnet}(\mathrm{x};\Theta_f^{*,(k+1)})$.
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, (2021).

Deep adaptive sampling method (DAS)

The framework of DAS. (see [Tang, Wan and Yang, 2021] 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, (2021).

Analysis of DAS

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

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

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

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

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

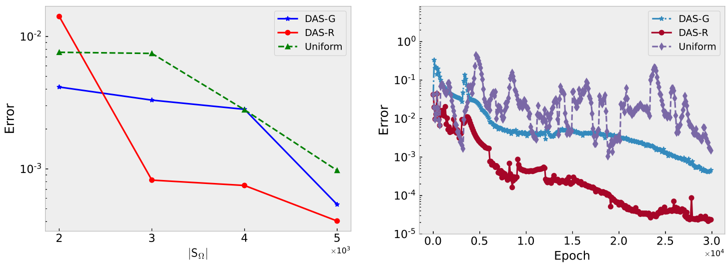
If the boundary loss $J_b(u)$ is zero, then the following inequality holds

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

Numerical experiments

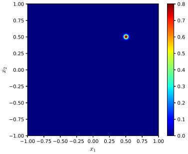
Two-dimensional peak problem

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

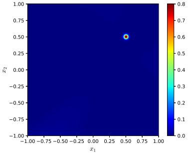


Numerical experiments

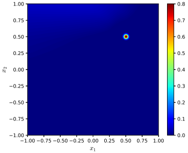
Two-dimensional peak problem



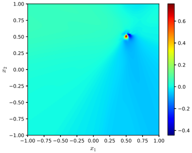
(a) The exact solution.



(b) DAS-R approximation.



(c) DAS-G approximation.

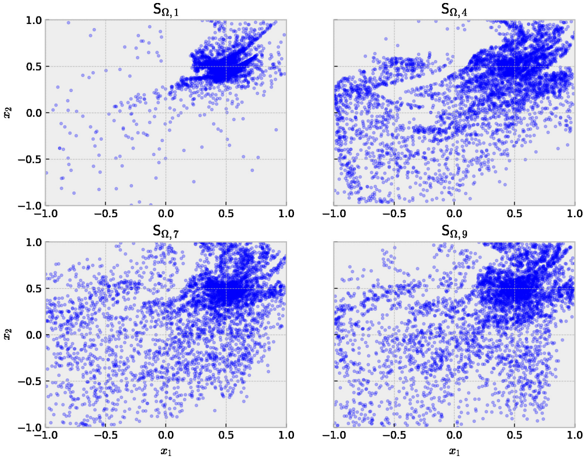


(d) Uniform sampling strategy.

Numerical experiments

Two-dimensional peak problem

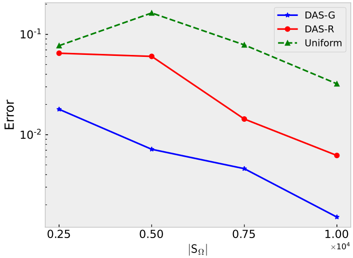
Samples

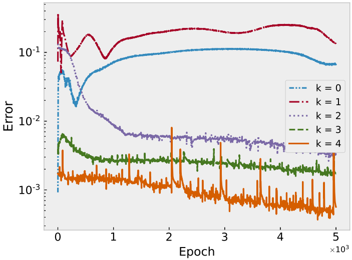


Numerical experiments

Two-dimensional problem with two peaks

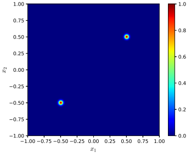
$$-\nabla \cdot \left[u(x_1,x_2)\nabla(x_1^2+x_2^2)\right] + \nabla^2 u(x_1,x_2) = s(x_1,x_2) \quad \text{in } \Omega,$$
$$u(x_1,x_2) = g(x_1,x_2) \quad \text{on } \partial\Omega,$$



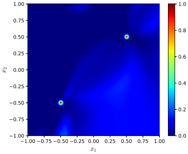


Numerical experiments

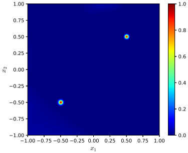
Two-dimensional problem with two peaks



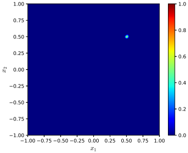
(e) The exact solution.



(f) DAS-R approximation.



(g) DAS-G approximation.

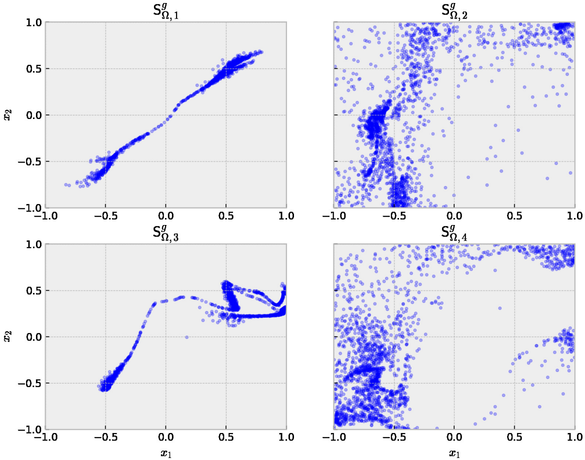


(h) Uniform sampling strategy.

Numerical experiments

Two-dimensional problem with two peaks

Samples



Numerical experiments

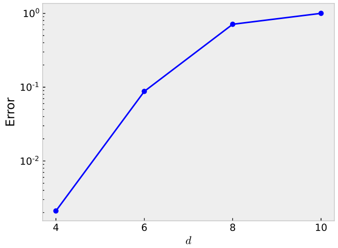
The d-dimensional elliptic equation

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

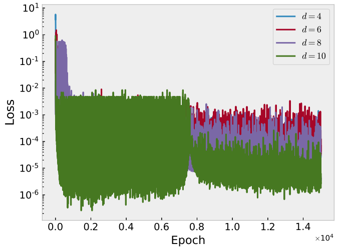
with an exact solution

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

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



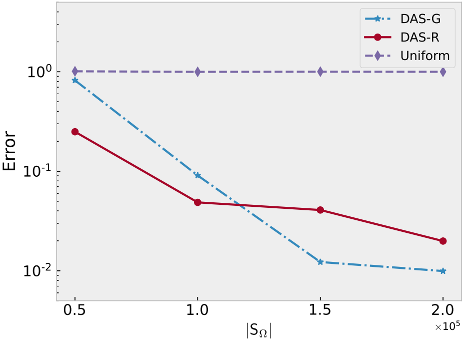
(i) Error



(j) Loss

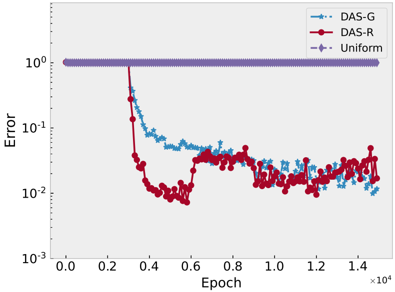
Numerical experiments

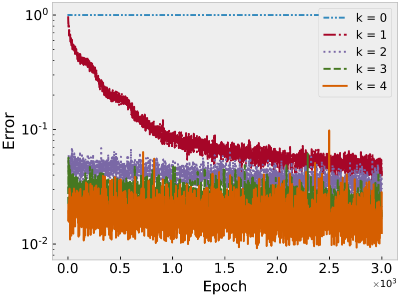
The 10-dimensional elliptic equation



Numerical experiments

The 10-dimensional elliptic equation

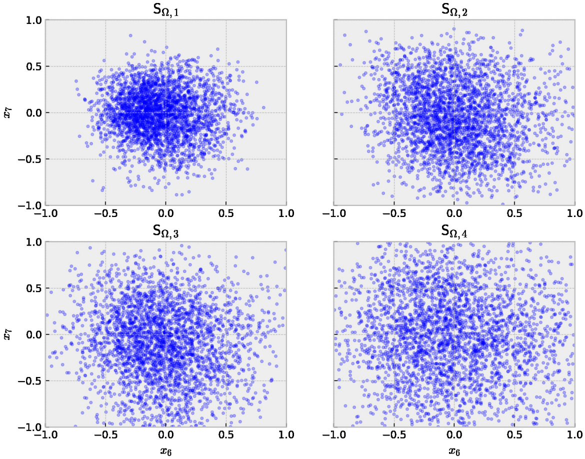




Numerical experiments

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