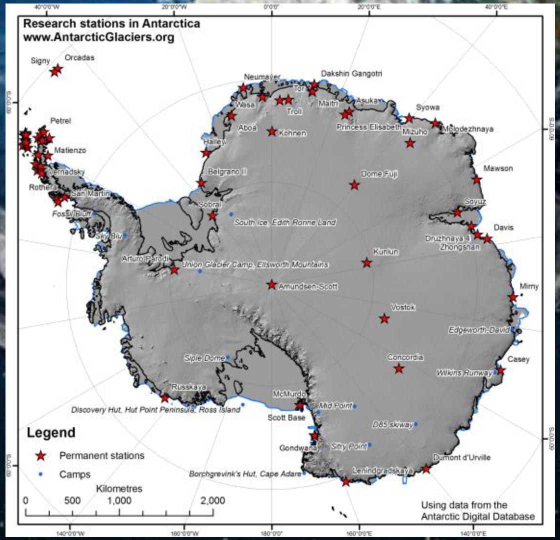
Deep Optimal Sensor Placement for Black Box Stochastic Simulations

University of Glasgow 28/05/2025

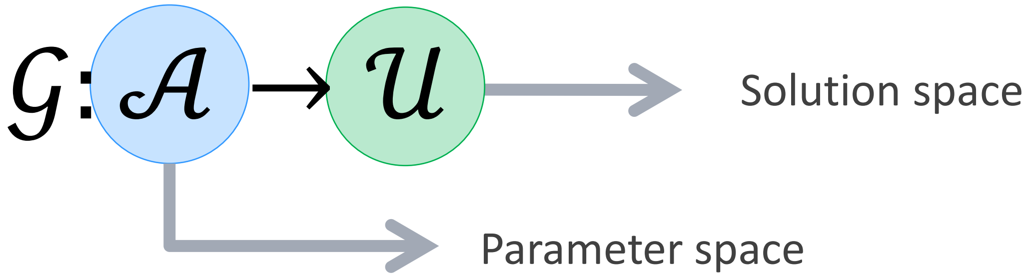
Paula Cordero Encinar, Tobias Schroder, Peter Yatsyshin and Andrew Duncan



Research stations in Antarctica
www.AntarcticGlaciers.org
Signy
Orcadas
Petrel
Matienzo
Vernadsky
Rothera
San Martin
Fossil Bluff
Sky Blu
Arturo Parodi
Neumayer
Dakshin Gangotri
Tor
Wasa
Troll
Maitri
Asuka
Syowa
Aboa
Kohnen
Princess Elisabeth
Mizuho
Molodezhnaya
Halley
Belgrano II
Dome Fuji
Mawson
Sobral
South Ice, Edith Ronne Land
Soyuz
Davis
Druzhnaya 4
Zhongshan
Union Glacier Camp, Ellsworth Mountains
Kunlun
Amundsen-Scott
Mirny
Vostok
Edgeworth-David
Siple Dome
Concordia
Casey
Wilkins Runway
Russkaya
McMurdo
Discovery Hut, Hut Point Peninsula, Ross Island
Mid Point
Scott Base
D85 skiway
Gondwana
Sitry Point
Dumont d'Urville
Borchgrevink's Hut, Cape Adare
Lenindgradskaya
Legend
Permanent stations
Camps
Kilometres
0 500 1,000 2,000
Using data from the
Antarctic Digital Database

PROBLEM FORMULATION

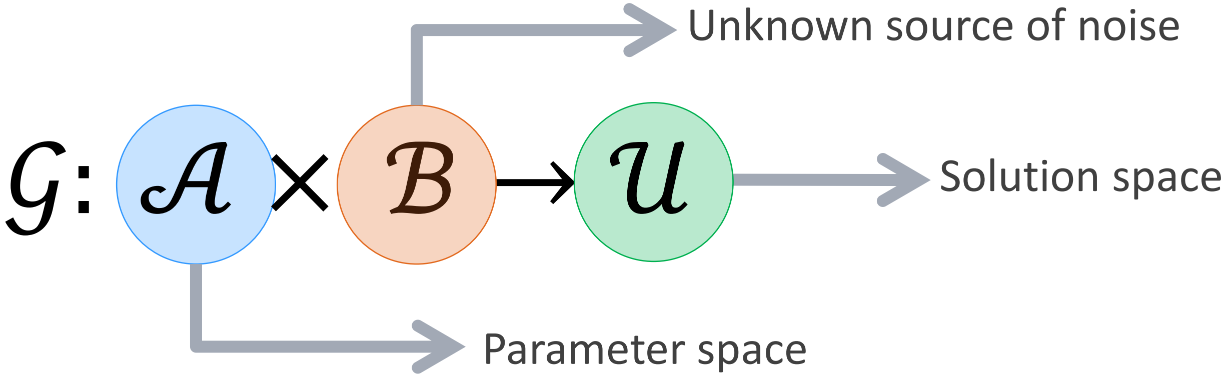
Consider a partial differential equation (PDE) model



Example: Poisson equation $\nabla u = a$. Given the functional parameter a, we compute the corresponding solution u.

PROBLEM FORMULATION

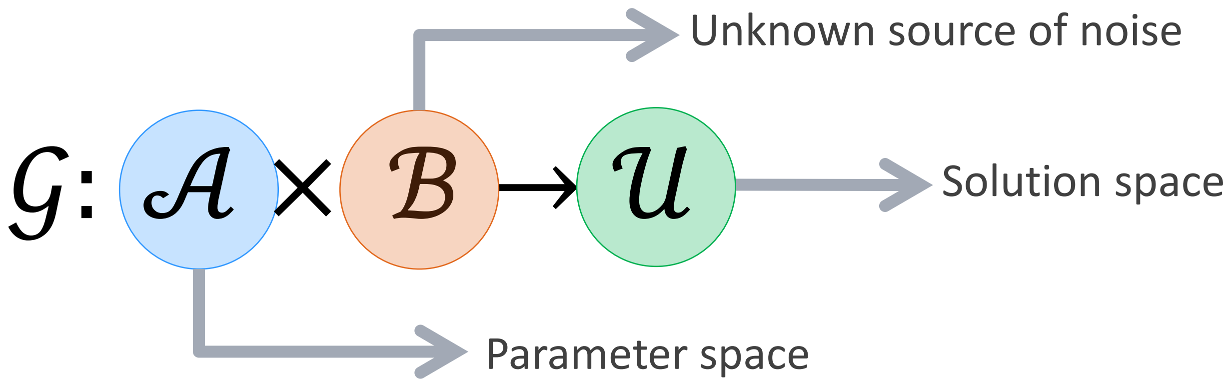
We go further and consider possibly stochastic PDEs



Example: Darcy flow $-\nabla \cdot (a(x)\nabla u(x)) = f(x) + \xi$, where ξ denotes space white noise. In this case, $u = \mathcal{G}_\xi(a)$.

PROBLEM FORMULATION

We will try to solve **inverse problems**



Given noisy observations of the sPDE solution $y_i = \mathcal{G}_\xi(a)(\boldsymbol{x}_i) + \eta_i$, **we want to infer $\boldsymbol{a}$**

In our scenario, η_i follows a standard Gaussian distribution. That is, $\eta_i \sim \mathcal{N}(0, \sigma^2)$.

PROBLEM FORMULATION

We will try to solve **inverse problems**

Unknown source of noise

$$\mathcal{G}: \mathcal{A} \times \mathcal{B} \rightarrow \mathcal{U}$$

Solution space

Parameter space

We take a Bayesian approach

Given noisy observations of the sPDE solution $y_i = \mathcal{G}_\xi(a)(\boldsymbol{x}_i) + \eta_i$, **we want to infer $\boldsymbol{a}$**

In our scenario, η_i follows a standard Gaussian distribution. That is, $\eta_i \sim \mathcal{N}(0, \sigma^2)$.

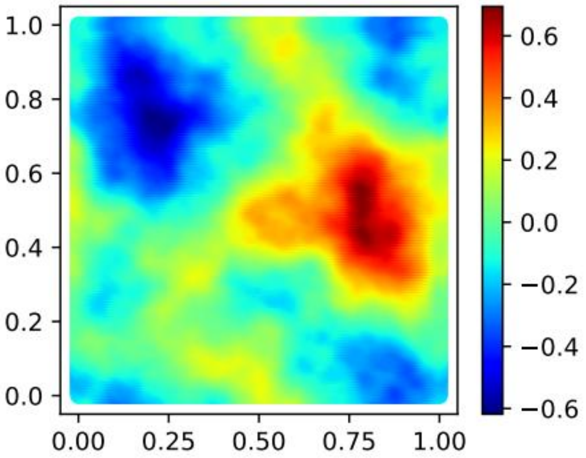
PROBLEM FORMULATION

An important challenge is sensor placement

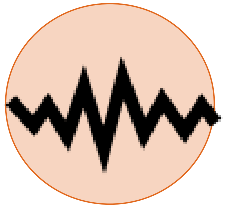
What does this mean? Determine measurement positions $\boldsymbol{x}$ that yield the most information about the solution u and the functional parameter a.

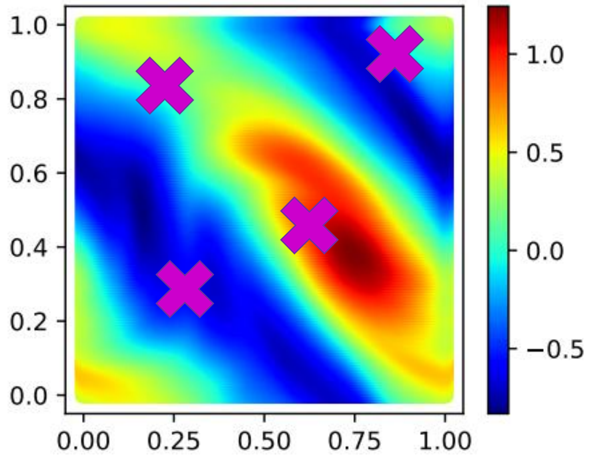
Recall that we measure noisy observations of the sPDE solution $y_i = \mathcal{G}_\xi(a)(\boldsymbol{x}_i) + \eta_i$ at different points $\boldsymbol{x}_i$ to infer the solution and the parameter a. We want to choose the sensor locations $\boldsymbol{x}_i$ in an optimal way.

OUR OBJECTIVE: learn the initial parameter + optimise sensor locations



×





Based on measurements of the solution by different sensors

How do we choose to obtain the maximum amount of information?

Infer the initial parameter

CHALLENGES

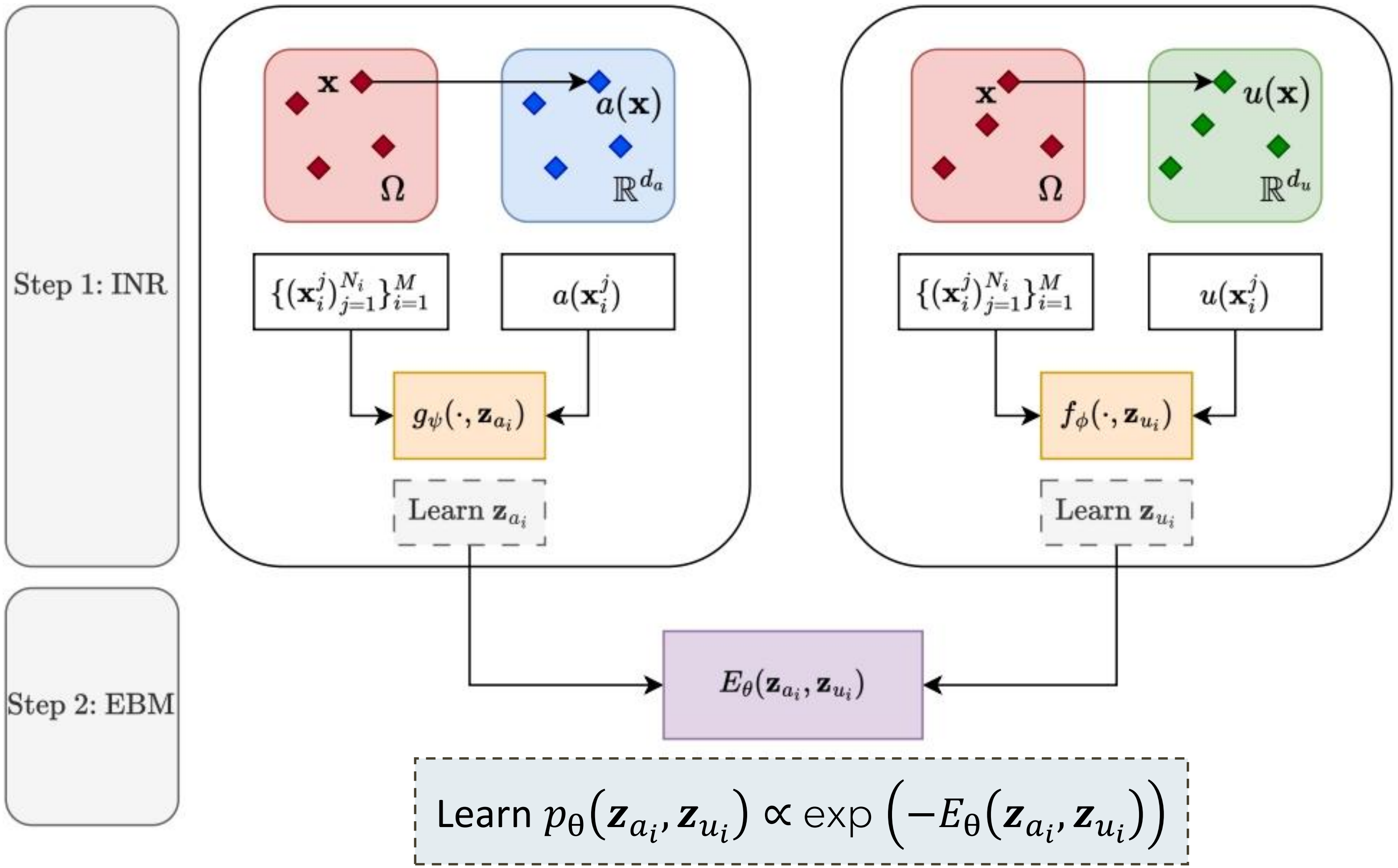
- Possibly stochastic operator $\mathcal{G}$
- Functional form parameter and solution
- Resolution invariant method
- Computationally efficient

SOLUTION

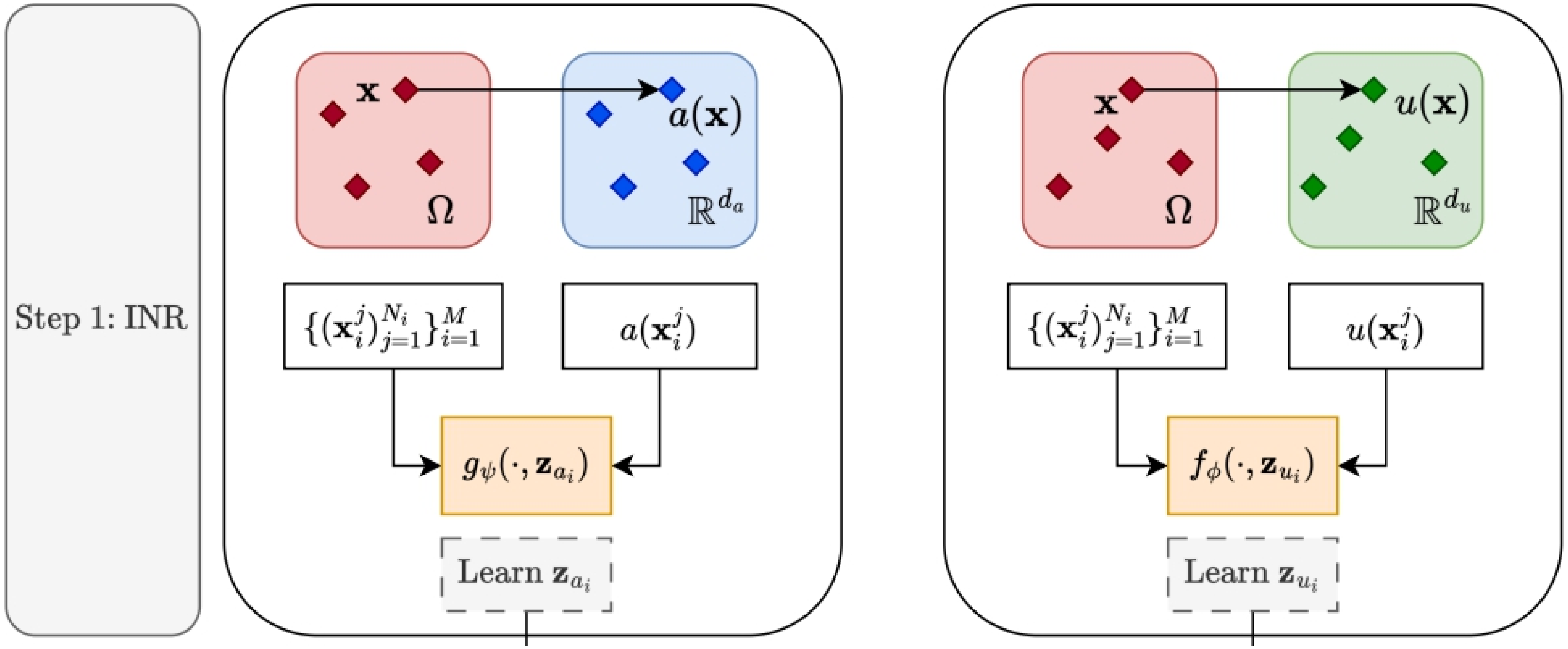
Learn a **generative model** $\boldsymbol{p_{\theta}}$ for the joint distribution over parameters and PDE solutions $(a\ , u = \mathcal{G}(a))$.

This provides a **surrogate method** which allows for complex probabilistic relationship and that is **not dependent on a fixed discretisation** of the domain.

TRAINING WORKFLOW



IMPLICIT NEURAL REPRESENTATIONS



- Shared weights among all functions in the dataset and particular weights for each function.
- Train them using an outer-inner optimisation loop to minimise the MSE.
- Very low reconstruction error.

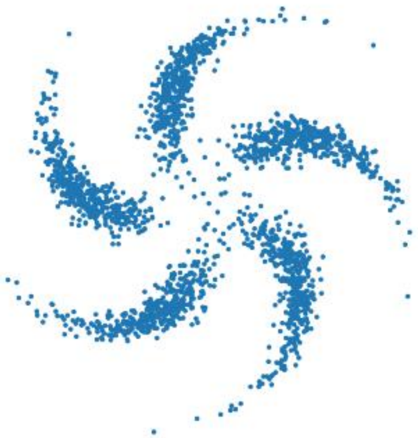
ENERGY-BASED MODELS

Learn $p_\theta(\boldsymbol{z}_{a_i}, \boldsymbol{z}_{u_i}) \propto \exp\left(-E_\theta(\boldsymbol{z}_{a_i}, \boldsymbol{z}_{u_i})\right)$

Energy Function (Neural Network)

Energy-Based Models (LeCun, 2006)

Data

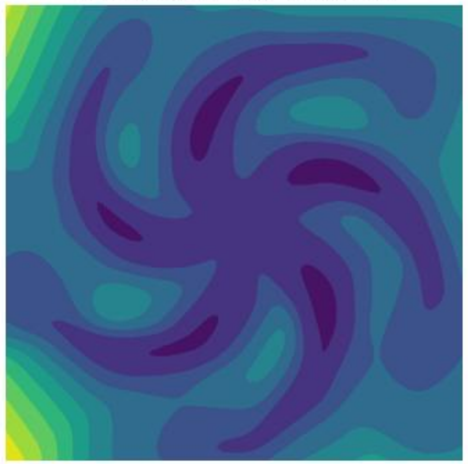


$\mathbf{x}^i \overset{i.i.d.}{\sim} p_{\text{data}}$

Model distribution

$$p_\theta(\mathbf{x}) = \frac{\exp(-U_\theta(\mathbf{x}))}{Z_\theta}$$

Learned Energy Function



$U_\theta(\mathbf{x})$

Intractable normalisation (partition function): $Z_\theta = \int \exp(-U_\theta(\mathbf{x}))\mathrm{d}\mathbf{x}$

ENERGY-BASED MODELS

Learn $p_\theta(\boldsymbol{z}_{a_i}, \boldsymbol{z}_{u_i}) \propto \exp\left(-E_\theta(\boldsymbol{z}_{a_i}, \boldsymbol{z}_{u_i})\right)$

- Training method:

Energy Discrepancy

Energy-Based distribution: $p(\mathbf{x}) \propto \exp(-U(\mathbf{x}))$

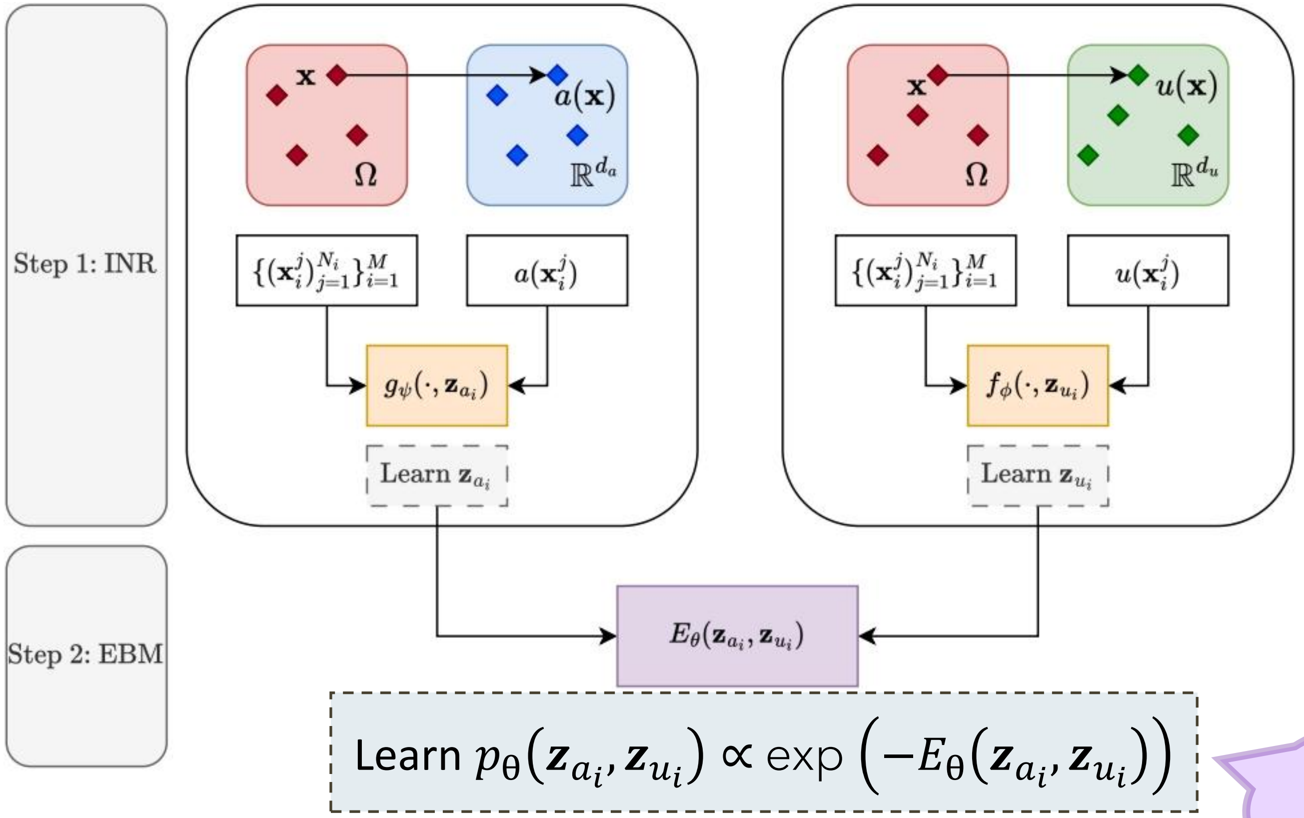
Conditional (noising) distribution: $q(\mathbf{y} \mid \mathbf{x})$

Contrastive potential: $U_q(\mathbf{y}) = -\log \int \exp(-U(\mathbf{x}))q(\mathbf{y} \mid \mathbf{x})\mathrm{d}\mathbf{x}$

$$\boxed{\mathrm{ED}_q(p_{\mathrm{data}}, p) := \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}[U(\mathbf{x})] - \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\mathbb{E}_{q(\mathbf{y}|\mathbf{x})}[U(\mathbf{y})]}$$

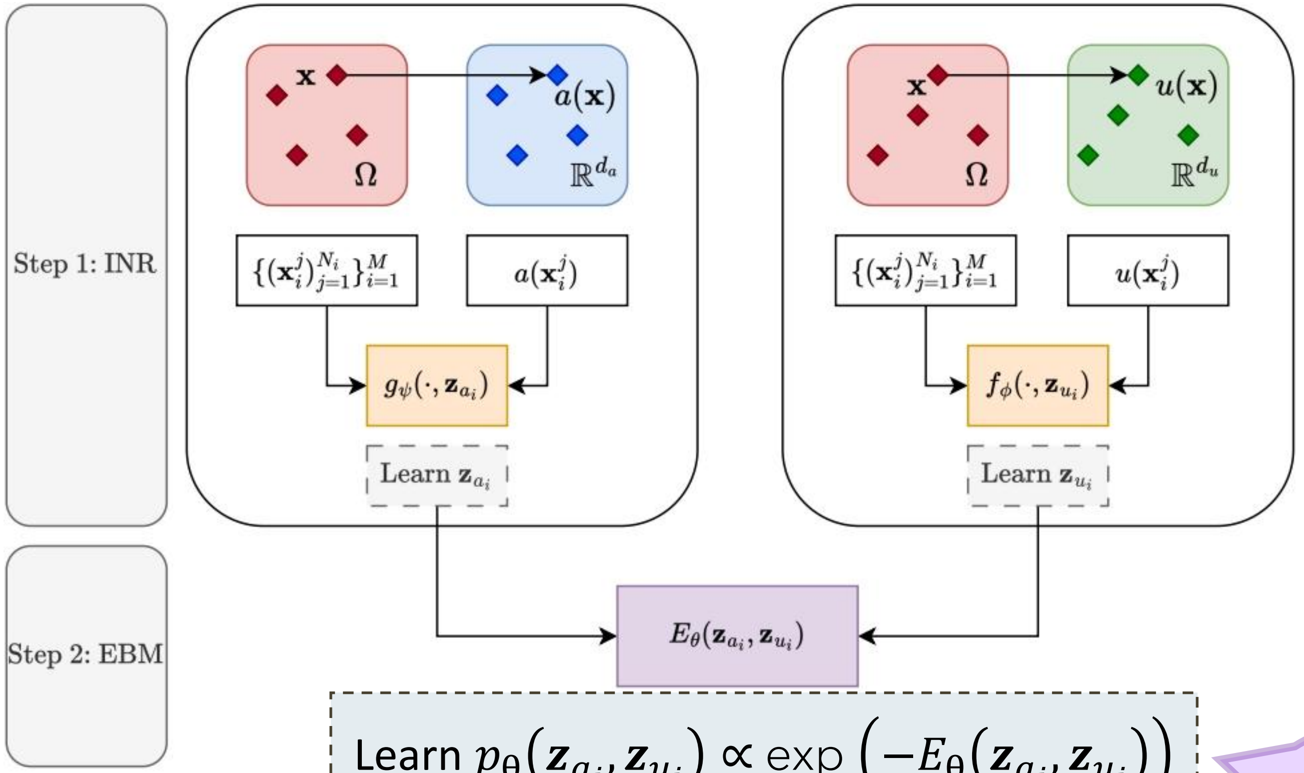
- Data processing inequality implies that $\mathrm{ED}_q(p_{data}, p) \geq 0$.
- Energy Discrepancy is functionally convex in U.
- For nice q, ED has a unique global minimiser at $\exp(-U^*) \propto p_{\mathrm{data}}$

INFERENCE FROM SPARSE OBSERVATIONS



Models distribution of latent representations

INFERENCE FROM SPARSE OBSERVATIONS

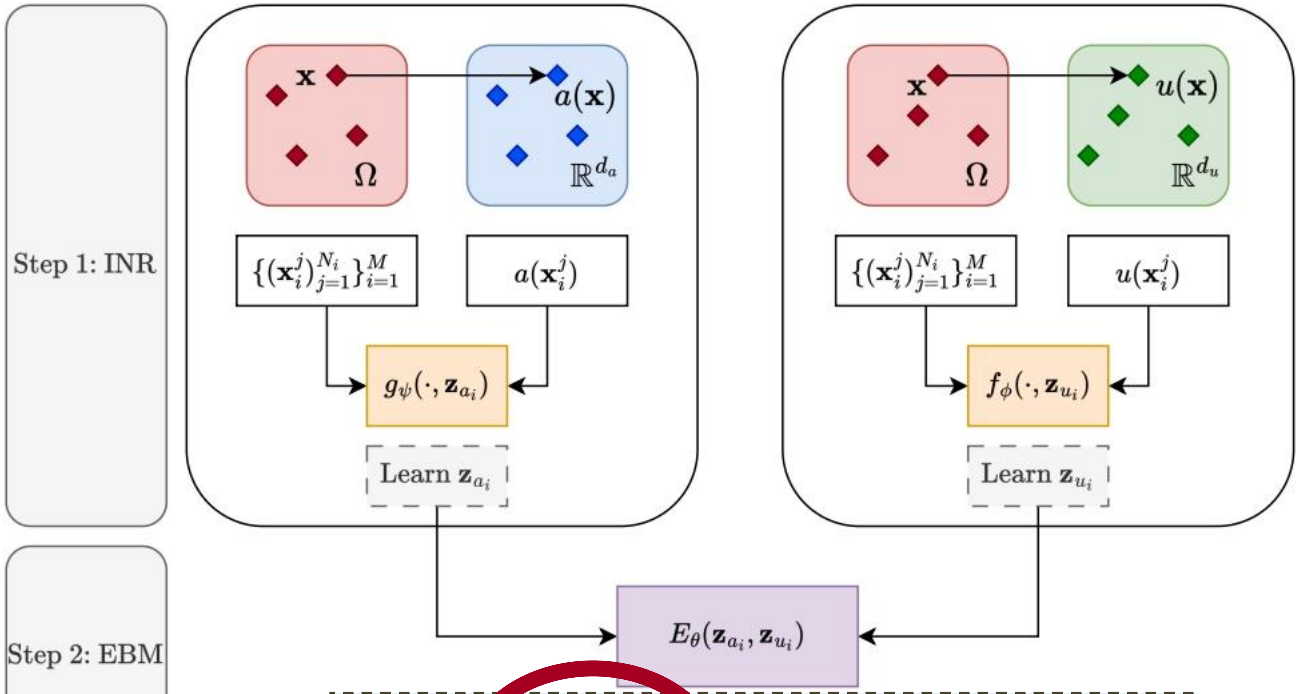


Learn $p_\theta(\mathbf{z}_{a_i}, \mathbf{z}_{u_i}) \propto \exp\left(-E_\theta(\mathbf{z}_{a_i}, \mathbf{z}_{u_i})\right)$

Models distribution of latent representations

It also provides a "prior" distribution for our problem

INFERENCE FROM SPARSE OBSERVATIONS



Learn $p_\theta(\boldsymbol{z}_{a_i}, \boldsymbol{z}_{u_i}) \propto \exp\left(-E_\theta(\boldsymbol{z}_{a_i}, \boldsymbol{z}_{u_i})\right)$

Allows uncertainty quantification

$$p(\boldsymbol{z}_a, \boldsymbol{z}_u | \mathcal{D}) \propto p(\mathcal{D} | \boldsymbol{z}_a, \boldsymbol{z}_u) p_\theta(\boldsymbol{z}_a, \boldsymbol{z}_u)$$

Bayes rule

OPTIMAL SENSOR PLACEMENT

Find optimal sparse sensor placement positions $\xi = \{\xi_1, \ldots, \xi_D\}$ to improve posterior inference based on new measurements $y_i = u(\xi_i) + \eta_i$.

HOW?

We need to define what is a good placement position, that is, an utility function.

OPTIMAL SENSOR PLACEMENT

Find optimal sparse sensor placement positions $\xi = \{\xi_1, \dots, \xi_D\}$ to improve posterior inference based on new measurements $y_i = u(\xi_i) + \eta_i$.

HOW?

Maximise utility of sensor placement positions

$$U(\xi) := \mathbb{E}_{p(y|\xi)} D_{KL}(p(\boldsymbol{z}_a, \boldsymbol{z}_u | y, \xi) || p_\theta(\boldsymbol{z}_a, \boldsymbol{z}_u))$$

OPTIMAL SENSOR PLACEMENT

Find optimal sparse sensor placement positions $\xi = \{\xi_1, \dots, \xi_D\}$ to improve posterior inference based on new measurements $y_i = u(\xi_i) + \eta_i$.

HOW?

$$U(\xi) := \mathbb{E}_{p(y|\xi)} D_{KL}(p(\boldsymbol{z}_a, \boldsymbol{z}_u | y, \xi) || p_\theta(\boldsymbol{z}_a, \boldsymbol{z}_u))$$

In practice, we maximise the PCE bound

$$\hat{U}_{PCE}(\xi) := \mathbb{E}\left[\log \frac{p(y|\boldsymbol{z}_{a,0}, \boldsymbol{z}_{u,0}, \xi)}{\frac{1}{L+1}\sum_{l=0}^{L} p(y|\boldsymbol{z}_{a,l}, \boldsymbol{z}_{u,l}, \xi)}\right] \leq U(\xi)$$

where the expectation is over $\prod p_\theta(\boldsymbol{z}_{a,i}, \boldsymbol{z}_{u,i})\, p(y|\boldsymbol{z}_{a,0}, \boldsymbol{z}_{u,0})$.

The selection of ξ_i is conducted sequentially.

OPTIMAL SENSOR PLACEMENT

Find optimal sparse sensor placement positions $\xi = \{\xi_1, \dots, \xi_D\}$ to improve posterior inference based on new measurements $y_i = u(\xi_i) + \eta_i$.

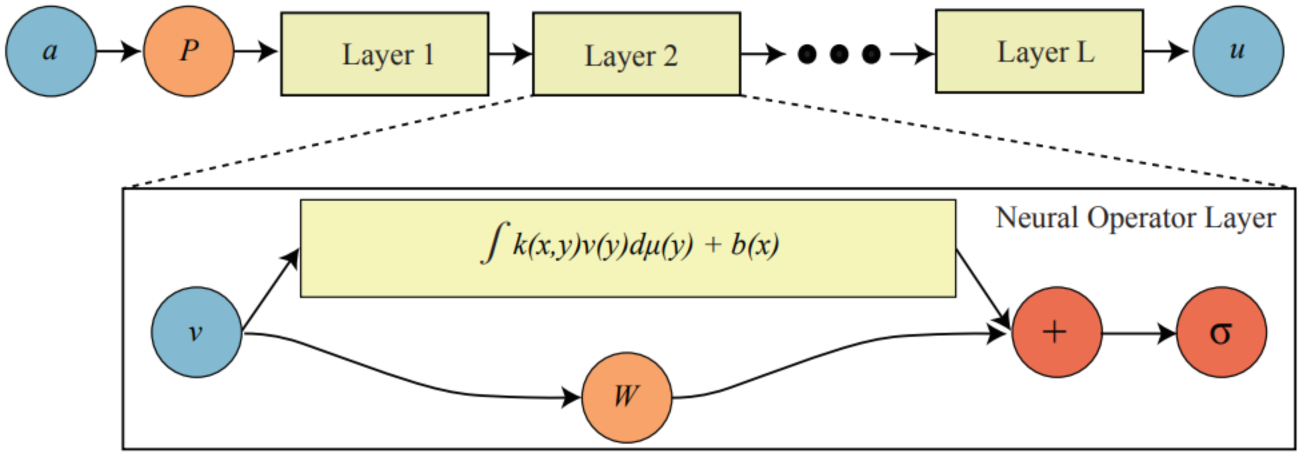
HOW TO DO THE SELECTION SEQUENTIALLY?

Considering the sequence of locations $\{\xi_1, \dots, \xi_{t-1}\}$ and outcomes $\{y_1, \dots, y_{t-1}\}$ up to step t, we maximise the utility given the history h_{t-1}

$$U(\xi_t | h_{t-1}) := \mathbb{E}\left[\log \frac{p(y | \mathbf{z}_a, \mathbf{z}_u, \xi_t, h_{t-1})}{p(y | \xi_t, h_{t-1})}\right]$$

BENCHMARK ALGORITHMS

- **Neural Operator Surrogate:** It can only learn deterministic maps. Therefore, it fails to incorporate the effect that a spatio-temporal external random signal has on the system described.



- **Neural Operator Surrogate With Noise Oracle** (Ideal setting not realistic): Takes as the driving noise of the stochastic model

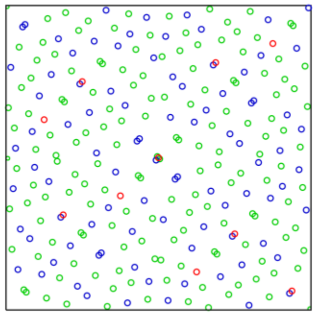
ALTERNATIVE METHODS

Methods	Direct PDE solves	Neural Operator surrogate	Neural Operator surrogate with oracle noise	Functional Neural Coupling (ours)
Low-cost evaluation	✗	✓	✓	✓
Low-cost inversion	✗	✓	✓	✓
Supports sensor placement pipeline	✓	✓	✓	✓
Supports stochastic PDEs	✓	✗	✓	✓
Tractable likelihood	✗	✗	✗	✓

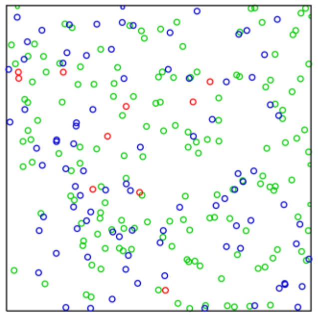
NUMERICAL EXAMPLES

- Our training data consists of M pairs of parameters and their corresponding solutions, $\{a_i,\ u_i\}_{i=1,\dots,M}$. We assume access to only N_i point observations for each of them, where the set of N_i locations varies across the M function realisations and need not be the same for a and u.
- PDE solutions are only required to train the INR and EBM models. Once trained, these models are reused for inference leading to high savings in terms of computational cost.
- For each method (ours and benchmarks) we compare optimal sensor placement against a quasi-Monte Carlo sequence

QMC



Uniform



NUMERICAL EXPERIMENTS

Boundary value problem in 1D: $u''(x) - u^2(x)u'(x) = f(x)$

Boundary conditions $u(-1) = X_a \sim N(a, 0.3^2), u(1) = X_b \sim Unif(b - 0.3, b + 0.4)$
Training data: a, b and observations of solution for a realisation of X_a, X_b

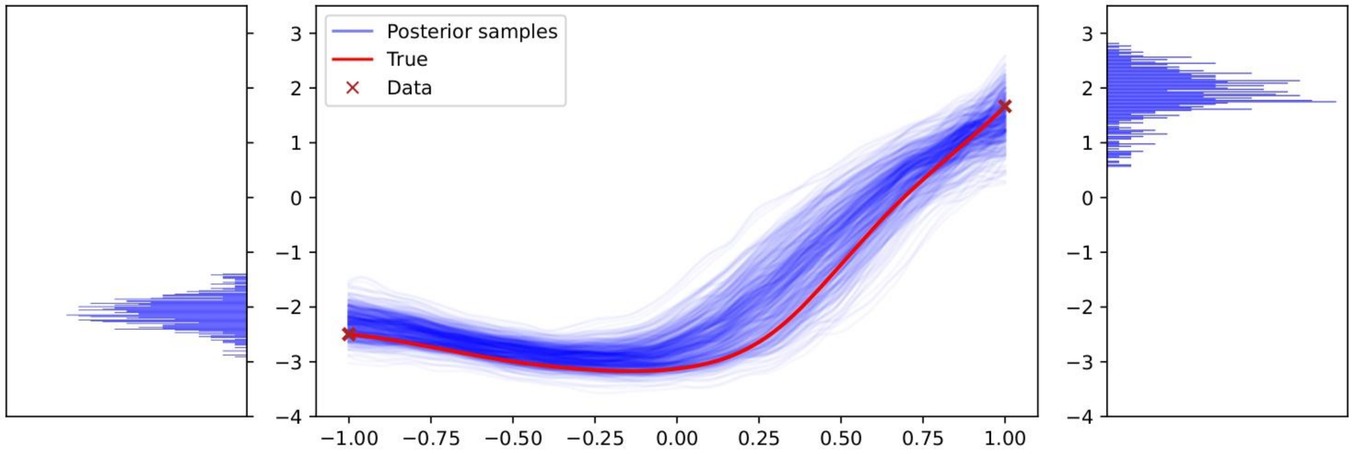
TO-DO

Perform inference on a, b based on 2 sparse observations of solution

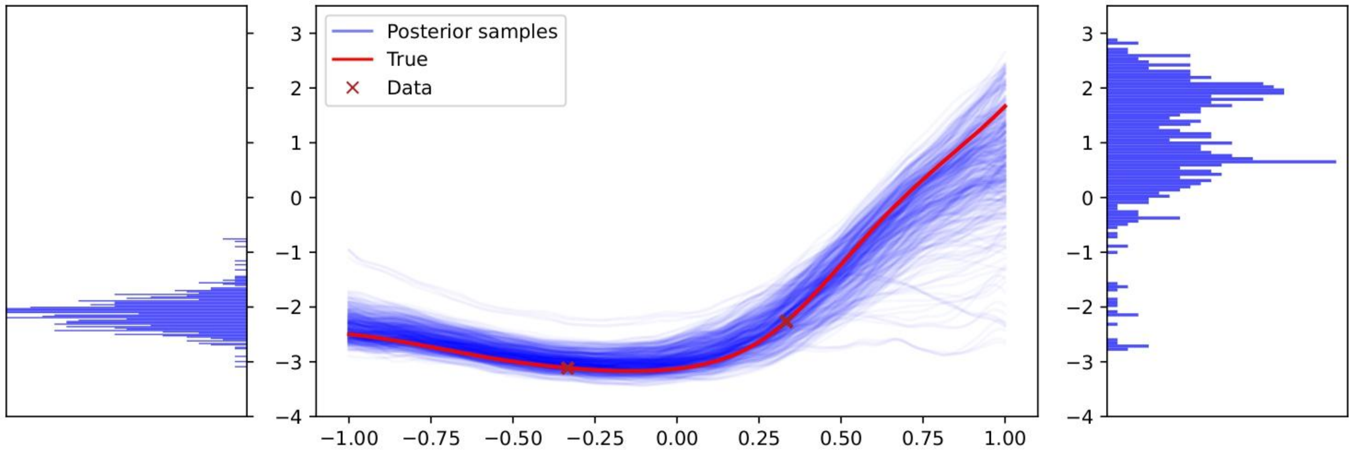
Method	Design points	$\|\widehat{u} - u_{\text{tr}}\|^2/\|u_{\text{tr}}\|^2$	MSE($\hat{a}$)	MSE($\hat{b}$)
Functional Neural Coupling (Ours)	Adaptive BED	0.124 ± 0.101	0.135 ± 0.099	1.441 ± 1.003
	Batch non-adaptive BED	$\mathbf{0.097 \pm 0.081}$	$\mathbf{0.103 \pm 0.193}$	$\mathbf{1.074 \pm 0.906}$
	Random sequence	0.331 ± 0.259	0.476 ± 0.888	4.162 ± 3.885
FNO surrogate (Li et al., 2021)	Adaptive BED	0.137 ± 0.308	0.441 ± 1.337	1.224 ± 2.084
	Batch non-adaptive BED	0.125 ± 0.255	0.428 ± 1.239	1.111 ± 1.785
	Random sequence	0.251 ± 0.507	0.484 ± 0.947	3.839 ± 7.457
FNO w/ oracle noise surrogate (Salvi et al., 2022)	Adaptive BED	0.116 ± 0.250	$\underline{0.021 \pm 0.058}$	1.580 ± 2.888
	Batch non-adaptive BED	$\underline{0.090 \pm 0.131}$	0.041 ± 0.105	$\underline{1.046 \pm 1.620}$
	Random sequence	0.356 ± 0.613	0.494 ± 1.412	8.372 ± 11.856

NUMERICAL EXPERIMENTS

Boundary value problem in 1D:



Batch non-adaptive



Sobol points

NUMERICAL EXPERIMENTS

Steady-state diffusion in 2D: $-\nabla \cdot \left(\kappa(x)\nabla u(x)\right) = f(x) + \alpha\omega$

Learn functional diffusion coefficient κ generated as the push-forward of a Gaussian random field. $f(x) = 0.5$ and ω is space white noise

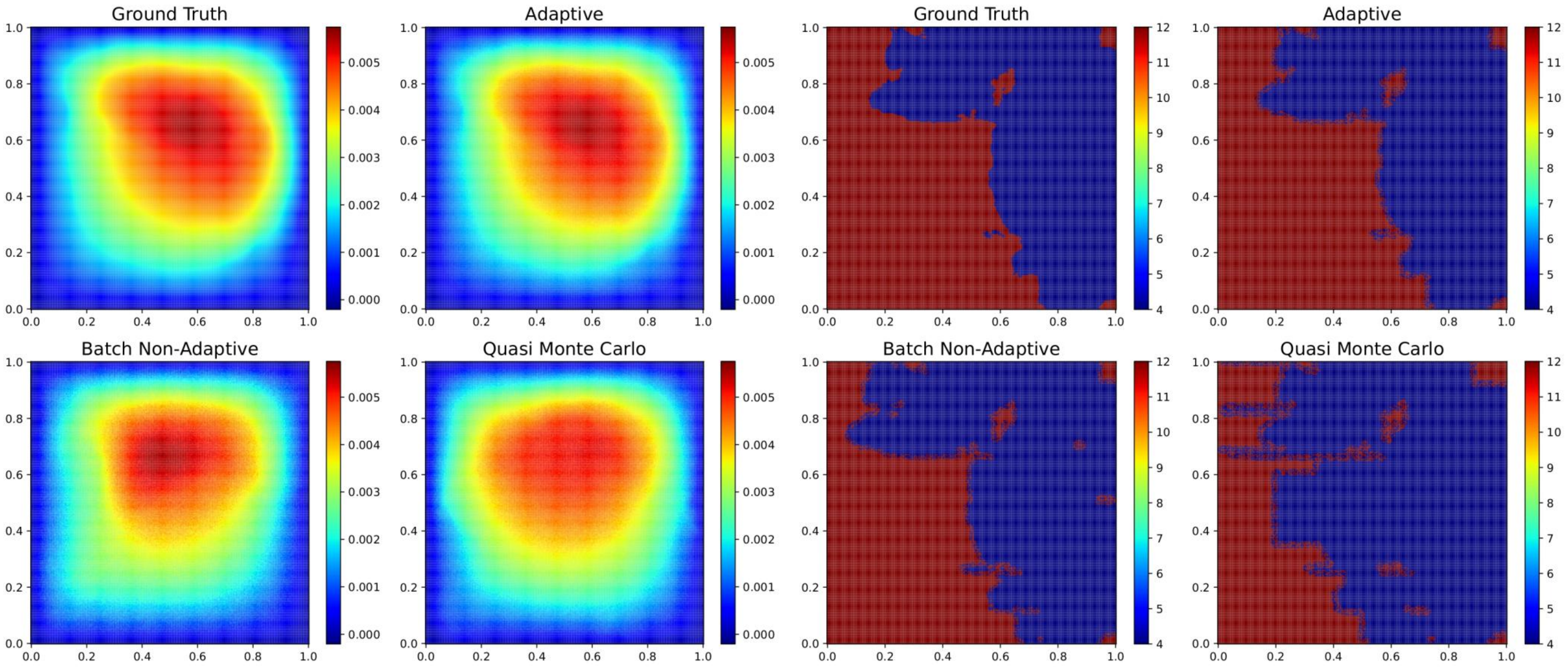
TO-DO

Based on 5 initial observations ✖ , find optimal locations ✖ for 15 additional measurement sites

Method	Design points	$\\|\log\widehat{\kappa} - \log\kappa_{\mathrm{tr}}\\|^2/\\|\log\kappa_{\mathrm{tr}}\\|^2$	$\\|\widehat{u} - u_{\mathrm{tr}}\\|^2/\\|u_{\mathrm{tr}}\\|^2$
Functional Neural Coupling (Ours)	Adaptive BED	$\mathbf{0.234 \pm 0.078}$	$\mathbf{0.102 \pm 0.083}$
	Batch non-adaptive BED	0.337 ± 0.091	0.192 ± 0.087
	Quasi-Monte Carlo sequence	0.711	0.328
FNO surrogate (Li et al., 2021)	Adaptive BED	0.306 ± 0.130	0.117 ± 0.106
	Batch non-adaptive BED	0.551 ± 0.172	0.220 ± 0.118
	Quasi-Monte Carlo sequence	1.182	0.379
FNO w/ oracle noise surrogate (Salvi et al., 2022)	Adaptive BED	$\underline{0.155 \pm 0.101}$	$\underline{0.093 \pm 0.089}$
	Batch non-adaptive BED	$\underline{0.291 \pm 0.089}$	$\underline{0.124 \pm 0.110}$
	Quasi-Monte Carlo sequence	0.459	0.255

NUMERICAL EXPERIMENTS

Steady-state diffusion in 2D:



Solutions

Diffusion coefficient

NUMERICAL EXPERIMENTS

Navier Stokes equation: $\partial_t \omega(x,t) + u(x,t) \cdot \nabla\omega(x,t) = \nu\Delta\omega(x,t) + f(x) + \alpha\varepsilon$

Learn initial vorticity $\omega_0(x) = \omega(x,0)$ generated according to a Gaussian random field with periodic boundary conditions. $f(x)$ is the deterministic forcing function and ε is the stochastic forcing function

TO-DO

We learn a Functional Neural Coupling between the initial vorticity ω_0 and the vorticity at times $t = 1, 2, 3$.
Find optimal locations for 15 measurements sites of the vorticity based on 5 initial observations.

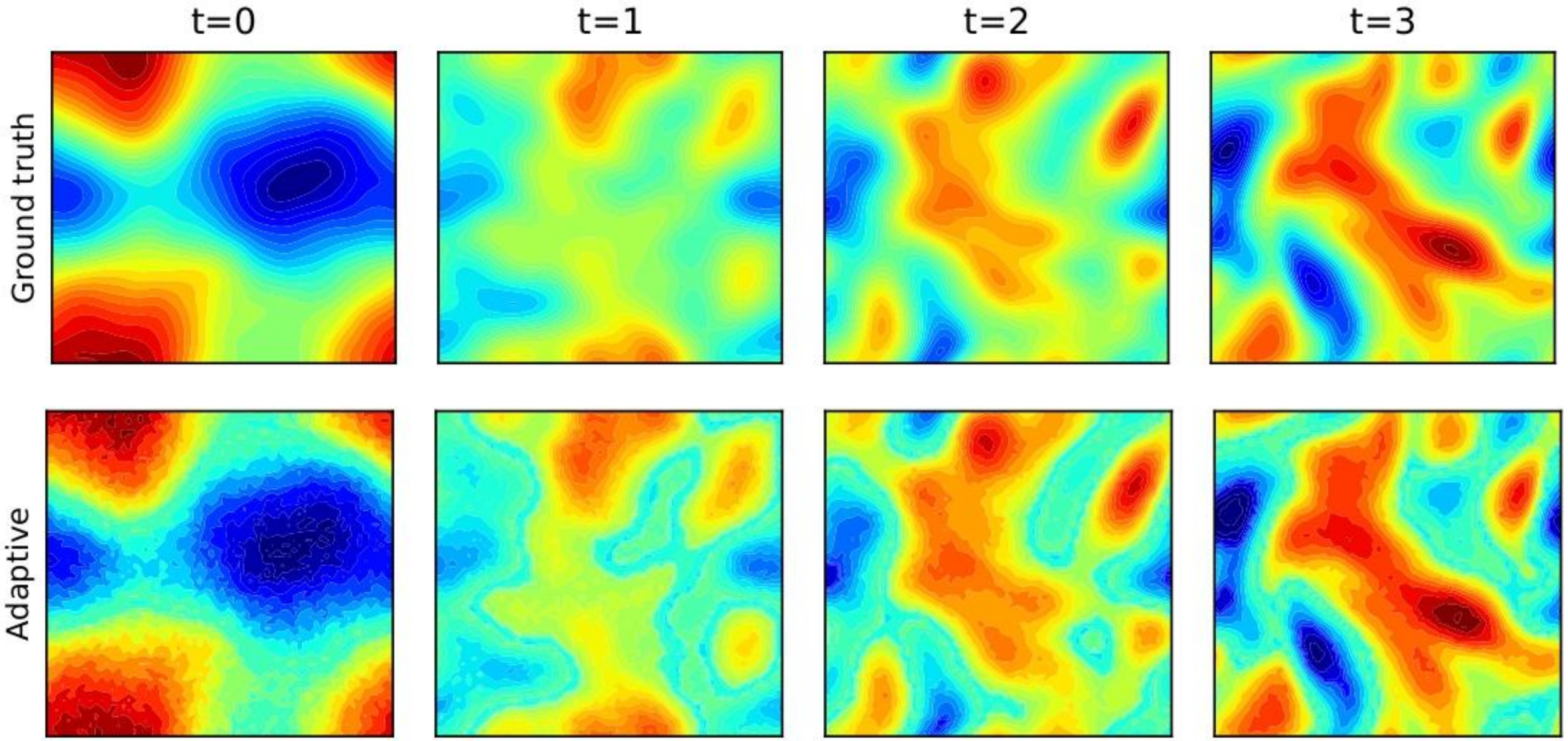
NUMERICAL EXPERIMENTS

Navier Stokes equation:

Method	Design points	$\|\widehat{w}_0 - w_{\mathrm{tr},0}\|^2 / \|w_{\mathrm{tr},0}\|^2$	$\|\underline{\widehat{w}}_t - \underline{w}_{\mathrm{tr},t}\|^2 / \|\underline{w}_{\mathrm{tr},t}\|^2$
	Adaptive BED	$\mathbf{0.293 \pm 0.077}$	$\mathbf{0.175 \pm 0.091}$
Functional Neural Coupling (Ours)	Batch non-adaptive BED	0.321 ± 0.083	0.239 ± 0.090
	Quasi-Monte Carlo sequence	0.578	0.422
FNO surrogate	Adaptive BED	0.382 ± 0.067	0.242 ± 0.095
(Li et al., 2021)	Batch non-adaptive BED	0.454 ± 0.092	0.288 ± 0.089
	Quasi-Monte Carlo sequence	0.652	0.576
FNO w/ oracle noise surrogate	Adaptive BED	$\underline{0.221 \pm 0.065}$	$\underline{0.103 \pm 0.079}$
(Salvi et al., 2022)	Batch non-adaptive BED	$\underline{0.301 \pm 0.080}$	$\underline{0.169 \pm 0.083}$
	Quasi-Monte Carlo sequence	0.454	0.332

NUMERICAL EXPERIMENTS

Navier Stokes equation:



LIMITATIONS

Limitations:

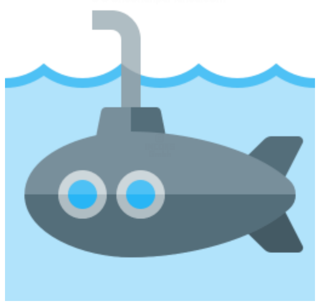
- Assumption that the density of the parameter-solution pairs is positive everywhere might be restrictive. For example, only a few parameter choices lead to stable behaviour of the system.
- EBM will not necessarily generalise to parts of the space not covered by the prior from which a was sampled. Therefore, we need to carefully choose training data.

Future work:

- Study sequential strategies that use the observation data more effectively for fine-tuning the base EBM. Generating more data as needed.

MORE FUTURE WORK

❓ If we have different sensors and some of them provide better measurements than others, how do we place them?



What if we have to select not a set of sensor points ✖️ but the route that a submarine follows to take measurements?

CONCLUSIONS

Our combination of implicit neural representation (INR) and generative model captures the often-intractable stochasticity that is propagated through the PDE and provides a **novel method for sensor placement in inverse PDE problems avoiding costly MCMC methods** with runtimes of days vs minutes for our approach.