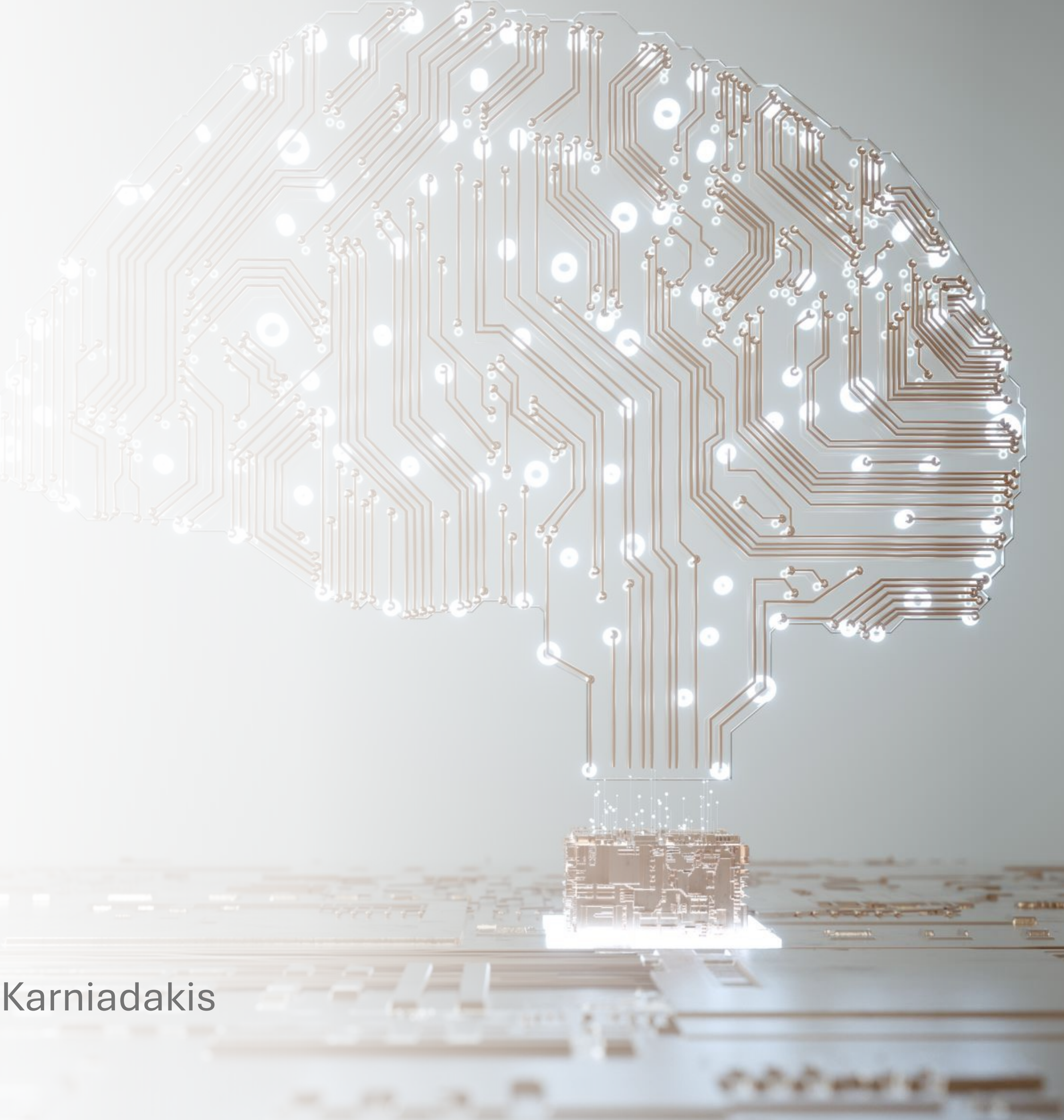
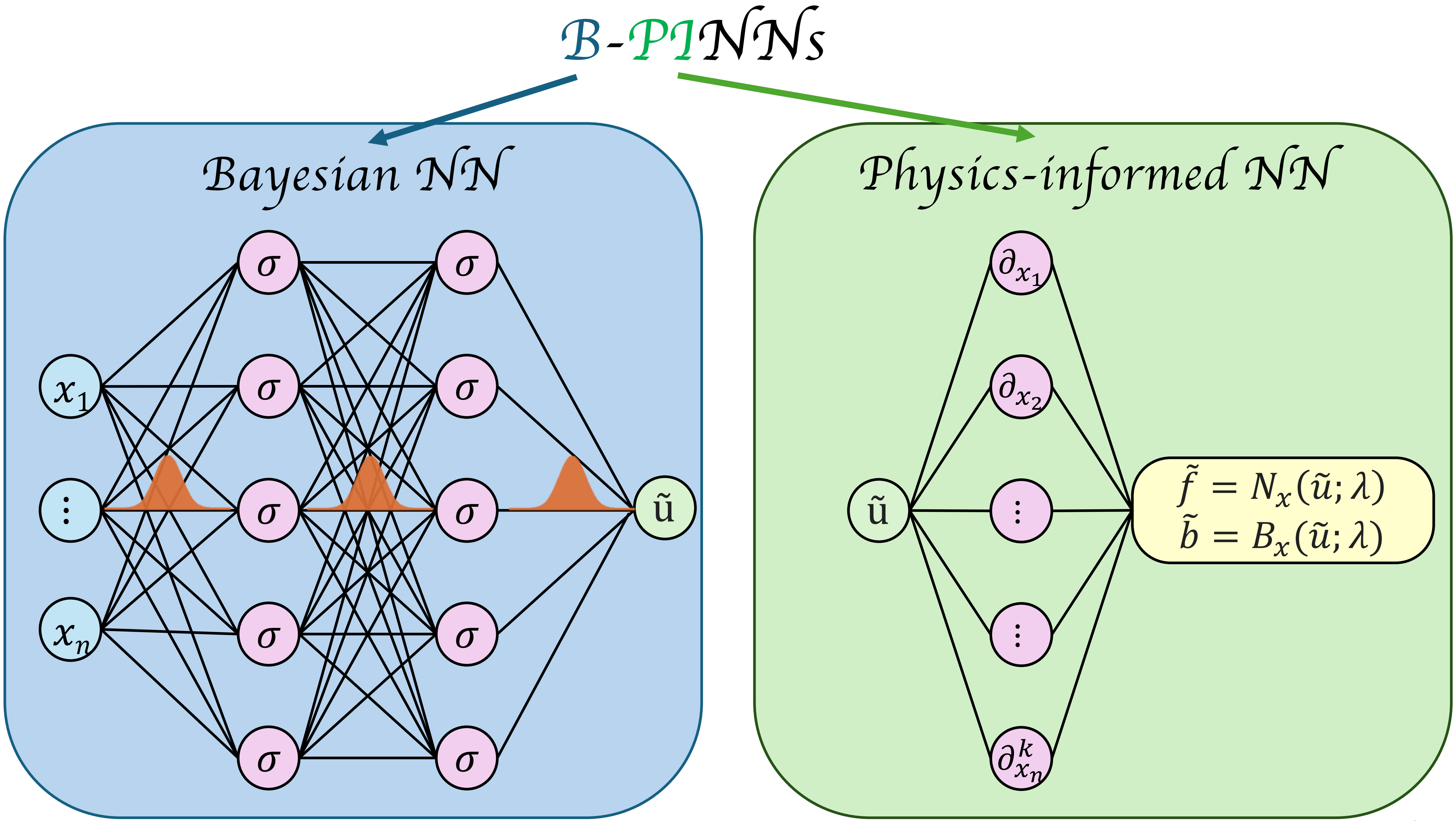
B-PINNs

"to solve linear and nonlinear PDEs with noisy data for both forward and inverse problems"



Paper by: Liu Yang, Xuhui Meng, George Em Karniadakis
Presented by: Sonja Joost, Gohar Tamrazyan



B-PINNs
Bayesian NN
Physics-informed NN
x_1
x_n
σ
ũ
∂_{x_1}
∂_{x_2}
$\partial_{x_n}^k$
$\tilde{f} = N_x(\tilde{u};\lambda)$
$\tilde{b} = B_x(\tilde{u};\lambda)$

Plan

- Motivation
- Framework
- Problem Approach
- Sampling Approaches
- Results

The Problem - Form of a general PDE

$$\mathcal{N}_{\boldsymbol{x}}(u;\lambda) = f, \qquad \boldsymbol{x} \in D$$

$$\mathcal{B}_{\boldsymbol{x}}(u;\lambda) = b, \qquad \boldsymbol{x} \in \Gamma$$

The Problem - Form of a general PDE

General differential operator

$$\mathcal{N}_{\boldsymbol{x}}(u;\lambda) = f, \qquad \boldsymbol{x} \in D$$

$$\mathcal{B}_{\boldsymbol{x}}(u;\lambda) = b, \qquad \boldsymbol{x} \in \Gamma$$

Boundary condition operator

The Problem - Form of a general PDE

General differential operator

Solution $u(\boldsymbol{x})$

$$\mathcal{N}_{\boldsymbol{x}}(u;\lambda) = f, \qquad \boldsymbol{x} \in D$$

$$\mathcal{B}_{\boldsymbol{x}}(u;\lambda) = b, \qquad \boldsymbol{x} \in \Gamma$$

Boundary condition operator

Vector of parameters

The Problem - Form of a general PDE

General differential operator

Solution $u(\boldsymbol{x})$

forcing terms $f(\boldsymbol{x}), b(\boldsymbol{x})$

$$\mathcal{N}_{\boldsymbol{x}}(u;\lambda) = \underline{f,} \qquad \boldsymbol{x} \in D$$

$$\mathcal{B}_{\boldsymbol{x}}(u;\lambda) = \underline{b,} \qquad \boldsymbol{x} \in \Gamma$$

Boundary condition operator

Vector of parameters

The Problem - Form of a general PDE

General differential operator

Solution $u(\boldsymbol{x})$

forcing terms $f(\boldsymbol{x}), b(\boldsymbol{x})$

Domain

$$\mathcal{N}_{\boldsymbol{x}}(u;\lambda) = f, \qquad \boldsymbol{x} \in \underline{D}$$

$$\mathcal{B}_{\boldsymbol{x}}(u;\lambda) = b, \qquad \boldsymbol{x} \in \underline{\Gamma}$$

Boundary condition operator

Vector of parameters

Domain boundary

Framework

What we have and what we are looking for.

Dataset $\mathcal{D}$

$$\mathcal{N}_{\boldsymbol{x}}(u;\lambda) = f, \qquad \boldsymbol{x} \in D$$
$$\mathcal{B}_{\boldsymbol{x}}(u;\lambda) = b, \qquad \boldsymbol{x} \in \Gamma$$

$$\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_f \cup \mathcal{D}_b$$

$$\mathcal{D}_u = \left\{\left(\boldsymbol{x}_u^{(i)}, \underline{\bar{u}^{(i)}}\right)\right\}_{i=1}^{N_u}$$

$$\mathcal{D}_f = \left\{\left(\boldsymbol{x}_f^{(i)}, \underline{\bar{f}^{(i)}}\right)\right\}_{i=1}^{N_f}$$

$$\mathcal{D}_b = \left\{\left(\boldsymbol{x}_b^{(i)}, \underline{\bar{b}^{(i)}}\right)\right\}_{i=1}^{N_b}$$

Dataset $\mathcal{D}$

$$\mathcal{N}_{\boldsymbol{x}}(u;\lambda) = f, \qquad \boldsymbol{x} \in D$$
$$\mathcal{B}_{\boldsymbol{x}}(u;\lambda) = b, \qquad \boldsymbol{x} \in \Gamma$$

$$\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_f \cup \mathcal{D}_b$$

$$\mathcal{D}_u = \left\{\left(\boldsymbol{x}_u^{(i)}, \bar{u}^{(i)}\right)\right\}_{i=1}^{N_u}$$

$$\mathcal{D}_f = \left\{\left(\boldsymbol{x}_f^{(i)}, \bar{f}^{(i)}\right)\right\}_{i=1}^{N_f}$$

$$\mathcal{D}_b = \left\{\left(\boldsymbol{x}_b^{(i)}, \bar{b}^{(i)}\right)\right\}_{i=1}^{N_b}$$

Measurements

$$\bar{u}^{(i)} = u\left(\boldsymbol{x}_u^{(i)}\right) + \epsilon_u^{(i)}, \qquad i = 1,2 \ldots N_u$$

$$\bar{f}^{(i)} = f\left(\boldsymbol{x}_f^{(i)}\right) + \epsilon_f^{(i)}, \qquad i = 1,2 \ldots N_f$$

$$\bar{b}^{(i)} = b\left(\boldsymbol{x}_b^{(i)}\right) + \epsilon_b^{(i)}, \qquad i = 1,2 \ldots N_b$$

Dataset $\mathcal{D}$

$$\mathcal{N}_{\boldsymbol{x}}(u;\lambda) = f, \qquad \boldsymbol{x} \in D$$
$$\mathcal{B}_{\boldsymbol{x}}(u;\lambda) = b, \qquad \boldsymbol{x} \in \Gamma$$

$$\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_f \cup \mathcal{D}_b$$

$$\mathcal{D}_u = \left\{\left(\boldsymbol{x}_u^{(i)}, \bar{u}^{(i)}\right)\right\}_{i=1}^{N_u}$$
$$\mathcal{D}_f = \left\{\left(\boldsymbol{x}_f^{(i)}, \bar{f}^{(i)}\right)\right\}_{i=1}^{N_f}$$
$$\mathcal{D}_b = \left\{\left(\boldsymbol{x}_b^{(i)}, \bar{b}^{(i)}\right)\right\}_{i=1}^{N_b}$$

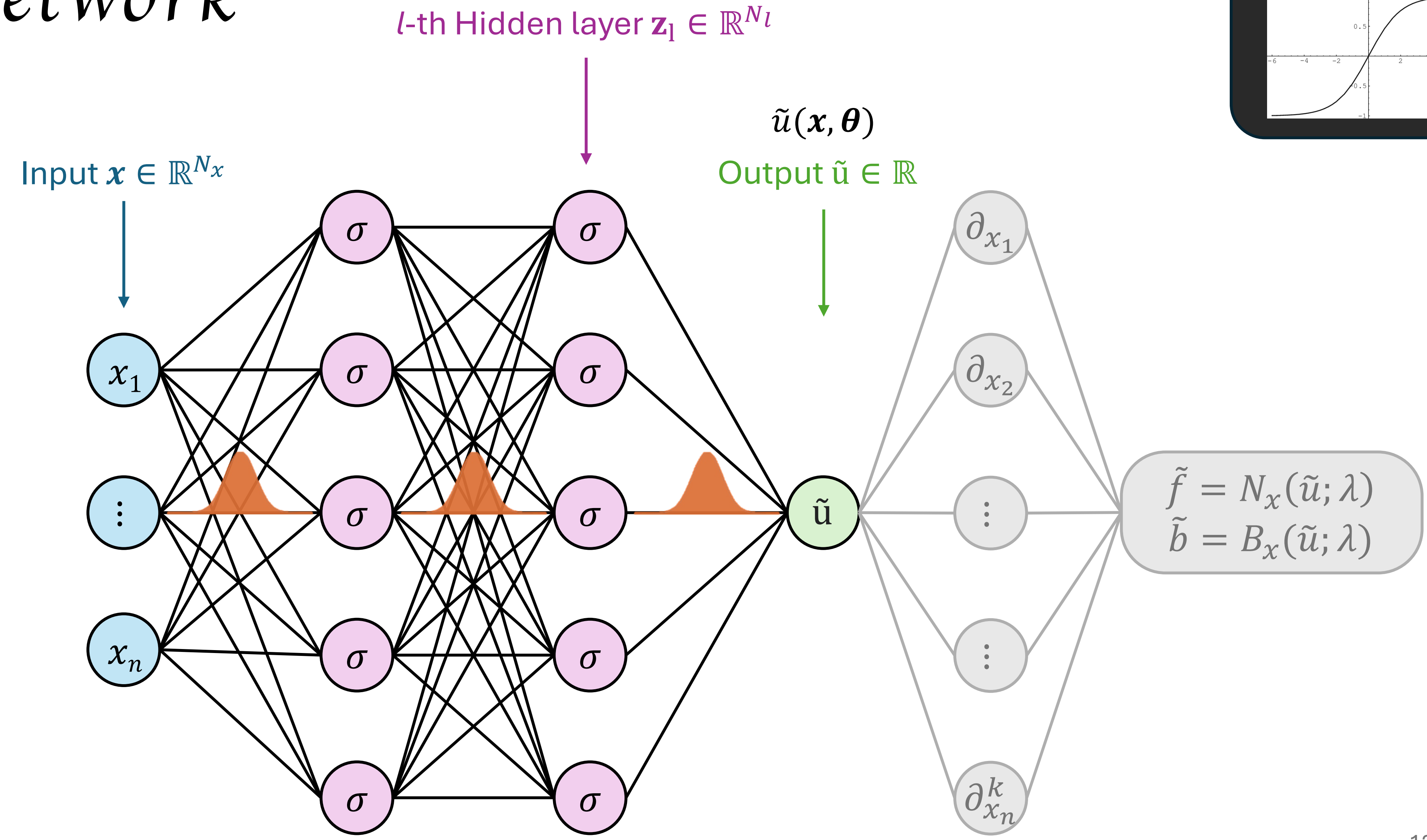
Measurements

Hidden real Values

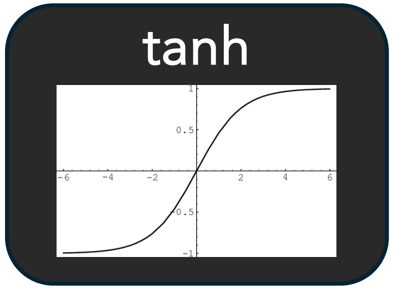
Independent Gaussian noises with zero mean

$$\bar{u}^{(i)} = u\left(\boldsymbol{x}_u^{(i)}\right) + \epsilon_u^{(i)}, \qquad i = 1,2 \ldots N_u$$
$$\bar{f}^{(i)} = f\left(\boldsymbol{x}_f^{(i)}\right) + \epsilon_f^{(i)}, \qquad i = 1,2 \ldots N_f$$
$$\bar{b}^{(i)} = b\left(\boldsymbol{x}_b^{(i)}\right) + \epsilon_b^{(i)}, \qquad i = 1,2 \ldots N_b$$

Network



l-th Hidden layer $\mathbf{z}_l \in \mathbb{R}^{N_l}$
$\tilde{u}(\boldsymbol{x}, \boldsymbol{\theta})$
Input $\boldsymbol{x} \in \mathbb{R}^{N_x}$
Output $\tilde{u} \in \mathbb{R}$
x_1
x_n
σ
$\tilde{u}$
∂_{x_1}
∂_{x_2}
$\partial^k_{x_n}$
$\tilde{f} = N_x(\tilde{u}; \lambda)$
$\tilde{b} = B_x(\tilde{u}; \lambda)$



tanh

Problem Approach

Problem approach

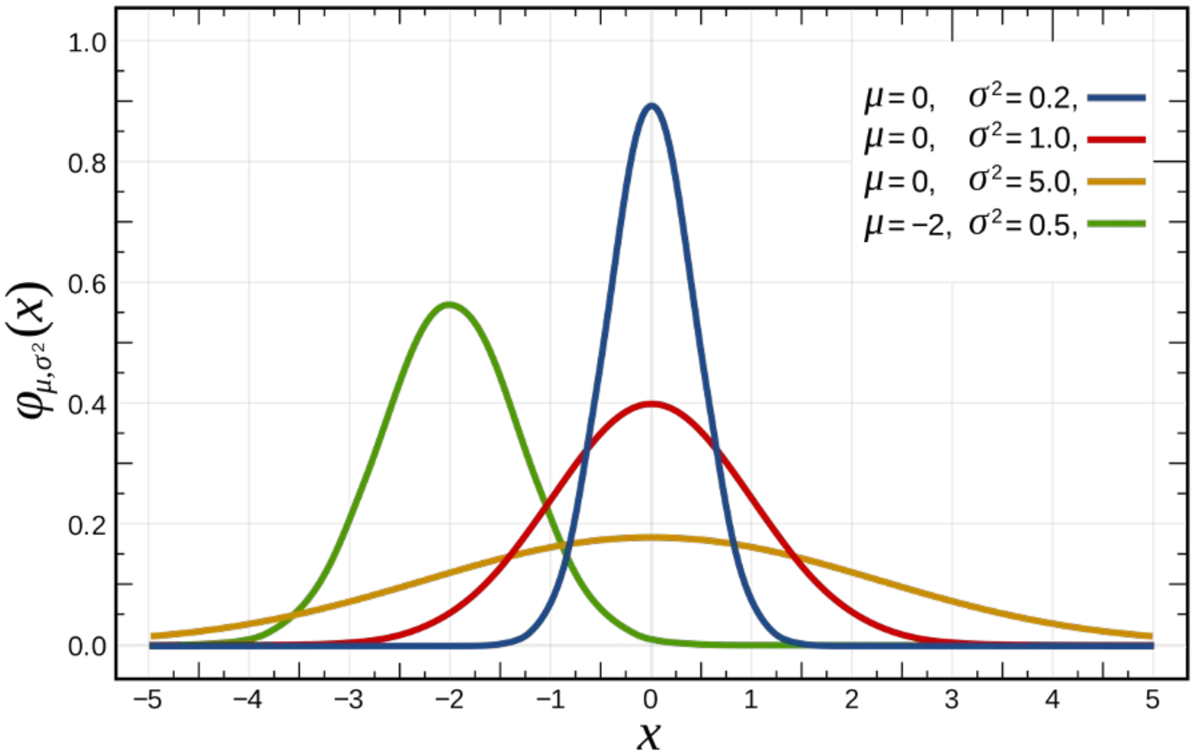
1. Assume Prior distribution $P(\boldsymbol{\theta})$
2. Compute Likelihood $P(\mathcal{D} \mid \boldsymbol{\theta})$
3. Compute Posterior Distribution $P(\boldsymbol{\theta} \mid \mathcal{D})$

Problem approach

1. Assume Prior distribution $P(\boldsymbol{\theta})$
2. Compute Likelihood $P(\mathcal{D} \mid \boldsymbol{\theta})$
3. Compute Posterior Distribution $P(\boldsymbol{\theta} \mid \mathcal{D})$
4. Sample from the Posterior Distribution $\{\boldsymbol{\theta}^{(i)}\}_{\{i=1\}}^{M}$
5. Obtain statistics from samples $\{\tilde{u}(\boldsymbol{x}, \boldsymbol{\theta}^{(i)})\}_{\{i=1\}}^{M}$

1. Prior Distribution

Gaussian Distributions with zero mean



2. Likelihood

$$P(\mathcal{D} \mid \boldsymbol{\theta}) = P\left(D_u \cup D_f \cup D_b \mid \boldsymbol{\theta}\right) = P(\mathcal{D}_u \mid \boldsymbol{\theta}) P\left(\mathcal{D}_f \mid \boldsymbol{\theta}\right) P(\mathcal{D}_b \mid \boldsymbol{\theta})$$

2. Likelihood

$$P(\mathcal{D} \mid \boldsymbol{\theta}) = P\left(D_u \cup D_f \cup D_b \mid \boldsymbol{\theta}\right) = P(\mathcal{D}_u \mid \boldsymbol{\theta})P\left(\mathcal{D}_f \mid \boldsymbol{\theta}\right)P(\mathcal{D}_b \mid \boldsymbol{\theta})$$

$$P(\mathcal{D}_u \mid \boldsymbol{\theta}) = P\left(\left\{\left(\boldsymbol{x}_u^{(i)}, \bar{u}^{(i)}\right)\right\}_{i=1}^{N_u} \mid \boldsymbol{\theta}\right) = \prod_{i=1}^{N_u} \frac{1}{\sqrt{2\pi {\sigma_u^{(i)}}^2}} \exp\left(-\frac{\left(\tilde{u}\left(\boldsymbol{x}_u^{(i)}; \boldsymbol{\theta}\right) - \bar{u}^{(i)}\right)^2}{2{\sigma_u^{(i)}}^2}\right)$$

2. Likelihood

Prior of $\boldsymbol{\theta}$ is a Gaussian Distribution

$$P(\mathcal{D} \mid \boldsymbol{\theta}) = P\left(D_u \cup D_f \cup D_b \mid \boldsymbol{\theta}\right) = P(\mathcal{D}_u \mid \boldsymbol{\theta}) P\left(\mathcal{D}_f \mid \boldsymbol{\theta}\right) P(\mathcal{D}_b \mid \boldsymbol{\theta})$$

$$P(\mathcal{D}_u \mid \boldsymbol{\theta}) = P\left(\left\{\left(\boldsymbol{x}_u^{(i)}, \bar{u}^{(i)}\right)\right\}_{i=1}^{N_u} \mid \boldsymbol{\theta}\right) = \prod_{i=1}^{N_u} \frac{1}{\sqrt{2\pi {\sigma_u^{(i)}}^2}} \exp\left(-\frac{\left(\tilde{u}\left(\boldsymbol{x}_u^{(i)}; \boldsymbol{\theta}\right) - \bar{u}^{(i)}\right)^2}{2{\sigma_u^{(i)}}^2}\right)$$

2. Likelihood

Computationally heavy

Prior of $\boldsymbol{\theta}$ is a Gaussian Distribution

$$P(\mathcal{D} \mid \boldsymbol{\theta}) = P\left(D_u \cup D_f \cup D_b \mid \boldsymbol{\theta}\right) = P(\mathcal{D}_u \mid \boldsymbol{\theta}) P\left(\mathcal{D}_f \mid \boldsymbol{\theta}\right) P(\mathcal{D}_b \mid \boldsymbol{\theta})$$

$$P(\mathcal{D}_u \mid \boldsymbol{\theta}) = P\left(\left\{\left(\boldsymbol{x}_u^{(i)}, \bar{u}^{(i)}\right)\right\}_{i=1}^{N_u} \mid \boldsymbol{\theta}\right) = \prod_{i=1}^{N_u} \frac{1}{\sqrt{2\pi {\sigma_u^{(i)}}^2}} \exp\left(-\frac{\left(\tilde{u}\left(\boldsymbol{x}_u^{(i)}; \boldsymbol{\theta}\right) - \bar{u}^{(i)}\right)^2}{2{\sigma_u^{(i)}}^2}\right)$$

$$P\left(\mathcal{D}_f \mid \boldsymbol{\theta}\right) = P\left(\left\{\left(\boldsymbol{x}_f^{(i)}, \bar{f}^{(i)}\right)\right\}_{i=1}^{N_f} \mid \boldsymbol{\theta}\right) = \prod_{i=1}^{N_f} \frac{1}{\sqrt{2\pi {\sigma_f^{(i)}}^2}} \exp\left(-\frac{\left(\tilde{f}\left(\boldsymbol{x}_f^{(i)}; \boldsymbol{\theta}\right) - \bar{f}^{(i)}\right)^2}{2{\sigma_f^{(i)}}^2}\right)$$

$$P(\mathcal{D}_b \mid \boldsymbol{\theta}) = P\left(\left\{\left(\boldsymbol{x}_b^{(i)}, \bar{b}^{(i)}\right)\right\}_{i=1}^{N_b} \mid \boldsymbol{\theta}\right) = \prod_{i=1}^{N_b} \frac{1}{\sqrt{2\pi {\sigma_b^{(i)}}^2}} \exp\left(-\frac{\left(\tilde{b}\left(\boldsymbol{x}_b^{(i)}; \boldsymbol{\theta}\right) - \bar{b}^{(i)}\right)^2}{2{\sigma_b^{(i)}}^2}\right)$$

3. Posterior Distribution

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})} \simeq P(\mathcal{D}|\theta)P(\theta) = \ likelihood \times prior$$

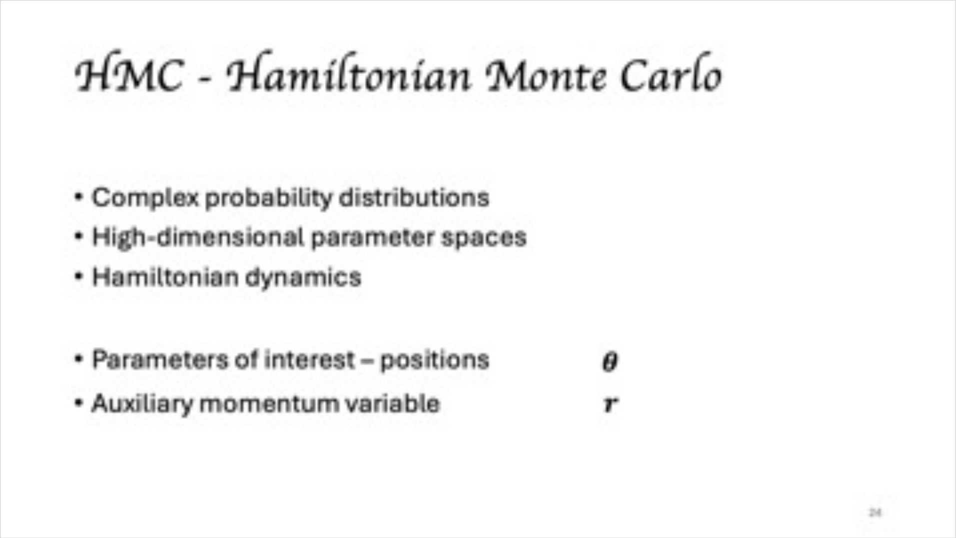
Bayes' Theorem

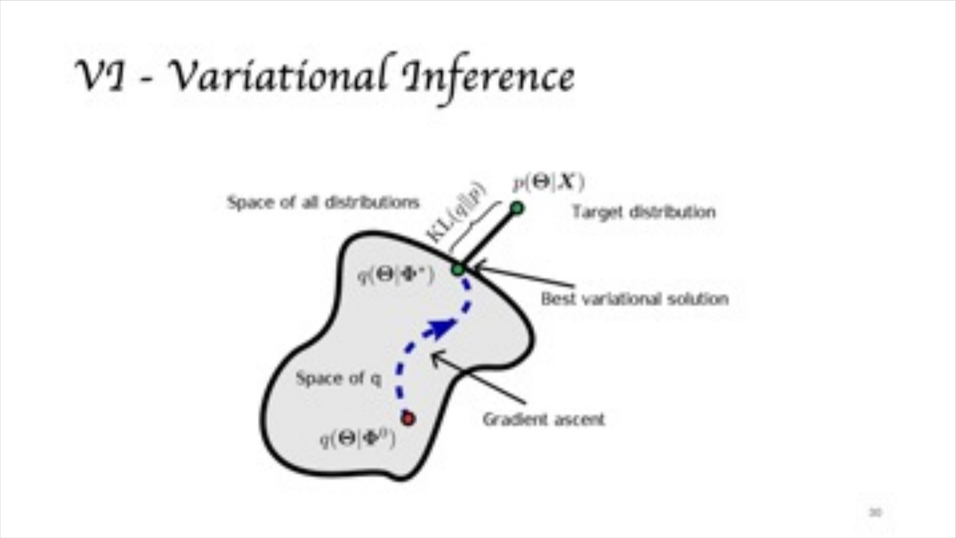
Equal up to a constant

Sampling Approaches

HMC and VI

4. Posterior Sampling Approaches





HMC - Hamiltonian Monte Carlo

- Complex probability distributions
- High-dimensional parameter spaces
- Hamiltonian dynamics

- Parameters of interest – positions $\boldsymbol{\theta}$
- Auxiliary momentum variable $\boldsymbol{r}$

HMC - Setup

$$P(\boldsymbol{\theta}|\mathcal{D}) \simeq P(\mathcal{D}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})$$

$$= \exp\Big(\ln\big(P(\mathcal{D}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})\big)\Big)$$

$$= \exp\Big(\ln\big(P(\mathcal{D}|\boldsymbol{\theta})\big) + \ln\big(P(\boldsymbol{\theta})\big)\Big)$$

$$= \exp\Big(-\boxed{\big(-\ln\big(P(\mathcal{D}|\boldsymbol{\theta})\big) - \ln\big(P(\boldsymbol{\theta})\big)\big)}\Big)$$

$$= \exp(-U(\boldsymbol{\theta}))$$

HMC - Setup

$$U(\boldsymbol{\theta}) = \left(-\ln\left(P(\mathcal{D}|\boldsymbol{\theta})\right) - \ln\left(P(\boldsymbol{\theta})\right)\right)$$

$$P(\theta|\mathcal{D}) \simeq \exp(-U(\boldsymbol{\theta}))$$

joint distribution $\boldsymbol{\pi}$:

$$\pi(\boldsymbol{\theta}, \boldsymbol{r}) \sim \exp\left(-U(\boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}^{-1}\boldsymbol{r}\right)$$

Hamiltonian system:

$$H(\boldsymbol{\theta}, \boldsymbol{r}) = U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}^{-1}\boldsymbol{r}$$

Hamiltonian dynamics:

$$\frac{d\boldsymbol{r}}{dt} = -\frac{\partial H}{\partial \boldsymbol{\theta}} \qquad \frac{d\boldsymbol{\theta}}{dt} = \frac{\partial H}{\partial \boldsymbol{r}}$$

$$d\boldsymbol{r} = -\nabla U(\boldsymbol{\theta})dt \qquad d\boldsymbol{\theta} = \boldsymbol{M}^{-1}\boldsymbol{r}dt$$

HMC - Setup

Hamiltonian dynamics:

$$\frac{d\boldsymbol{r}}{dt} = -\frac{\partial H}{\partial \boldsymbol{\theta}} \qquad \frac{d\boldsymbol{\theta}}{dt} = \frac{\partial H}{\partial \boldsymbol{r}}$$

$$d\boldsymbol{r} = -\nabla U(\boldsymbol{\theta})dt \qquad d\boldsymbol{\theta} = \boldsymbol{M}^{-1}\boldsymbol{r}dt$$

$$\frac{d}{dt}H\big(\boldsymbol{\theta}(t), \boldsymbol{r}(t)\big)$$

$$= \frac{\partial H}{\partial \boldsymbol{\theta}} \cdot \frac{d\boldsymbol{\theta}}{dt} + \frac{\partial H}{\partial \boldsymbol{r}} \cdot \frac{d\boldsymbol{r}}{dt}$$

$$= -\frac{d\boldsymbol{r}}{dt} \cdot \frac{d\boldsymbol{\theta}}{dt} + \frac{d\boldsymbol{\theta}}{dt} \cdot \frac{d\boldsymbol{r}}{dt}$$

$$= 0$$

HMC

Leapfrog integration

Metropolis Hastings

Algorithm 1 Hamiltonian Monte Carlo.

Require: initial states for $\boldsymbol{\theta}^{t_0}$ and time step size δt.

for $k = 1, 2 \ldots N$ **do**

Sample $\boldsymbol{r}^{t_{k-1}}$ from $\mathcal{N}(0, \boldsymbol{M})$,

$(\boldsymbol{\theta}_0, \boldsymbol{r}_0) \leftarrow (\boldsymbol{\theta}^{t_{k-1}}, \boldsymbol{r}^{t_{k-1}})$.

for $i = 0, 1 \ldots (L-1)$ **do**

$\boldsymbol{r}_i \leftarrow \boldsymbol{r}_i - \frac{\delta t}{2} \nabla U(\boldsymbol{\theta}_i)$,

$\boldsymbol{\theta}_{i+1} \leftarrow \boldsymbol{\theta}_i + \delta t \boldsymbol{M}^{-1} \boldsymbol{r}_i$,

$\boldsymbol{r}_{i+1} \leftarrow \boldsymbol{r}_i - \frac{\delta t}{2} \nabla U(\boldsymbol{\theta}_{i+1})$,

end for

Metropolis-Hastings step:

Sample p from Uniform[0, 1],

$\alpha \leftarrow \min\{1, \exp(H(\boldsymbol{\theta}_L, \boldsymbol{r}_L) - H(\boldsymbol{\theta}^{t_{k-1}}, \boldsymbol{r}^{t_{k-1}}))\}$.

if $p \geq \alpha$ **then**

$\boldsymbol{\theta}^{t_k} \leftarrow \boldsymbol{\theta}_L$,

else

$\boldsymbol{\theta}^{t_k} \leftarrow \boldsymbol{\theta}^{t_{k-1}}$.

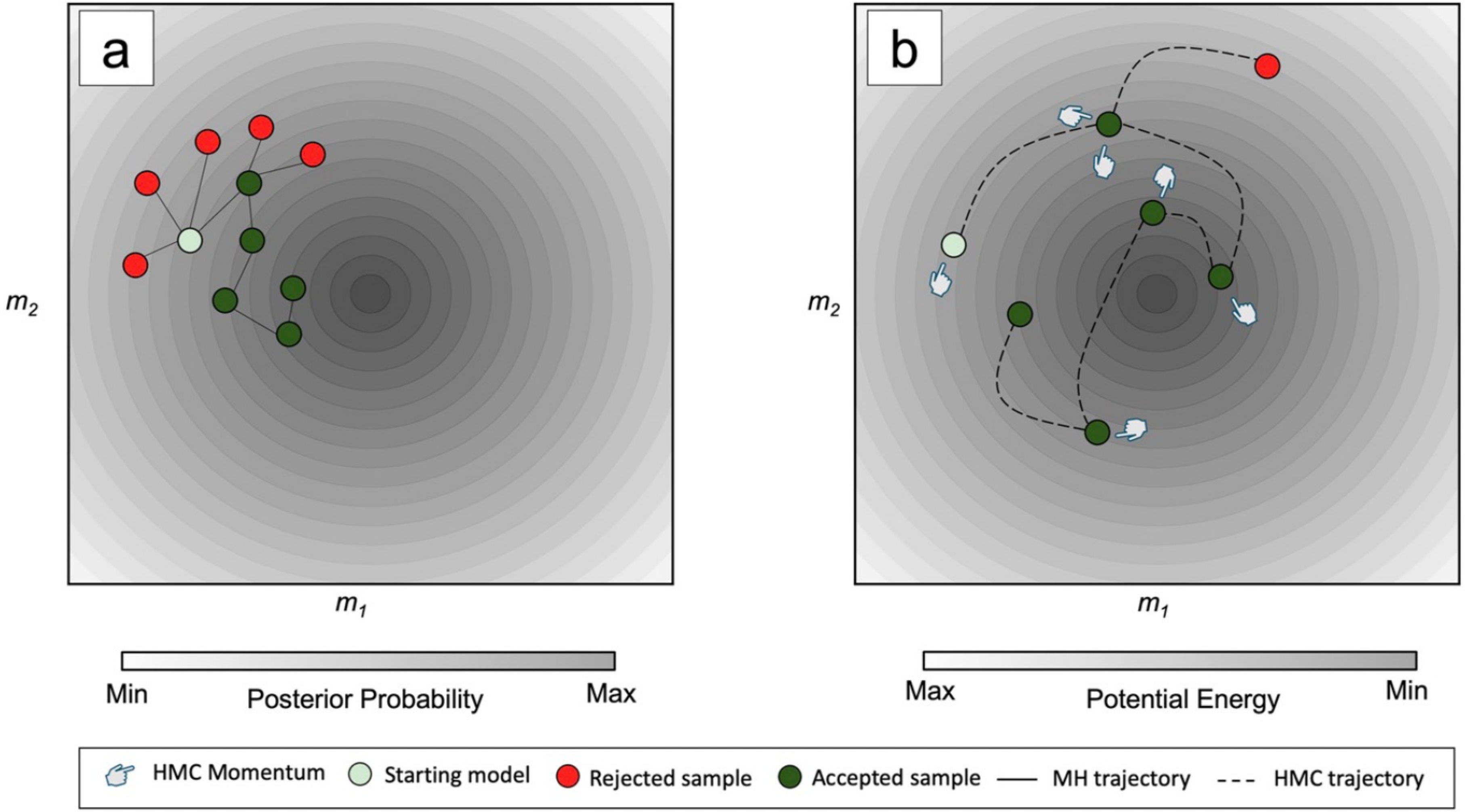
end if

end for

Calculate $\{\tilde{u}(\boldsymbol{x}, \boldsymbol{\theta}^{t_{N+1-j}})\}_{j=1}^{M}$ as samples of $u(\boldsymbol{x})$, similarly for other terms.

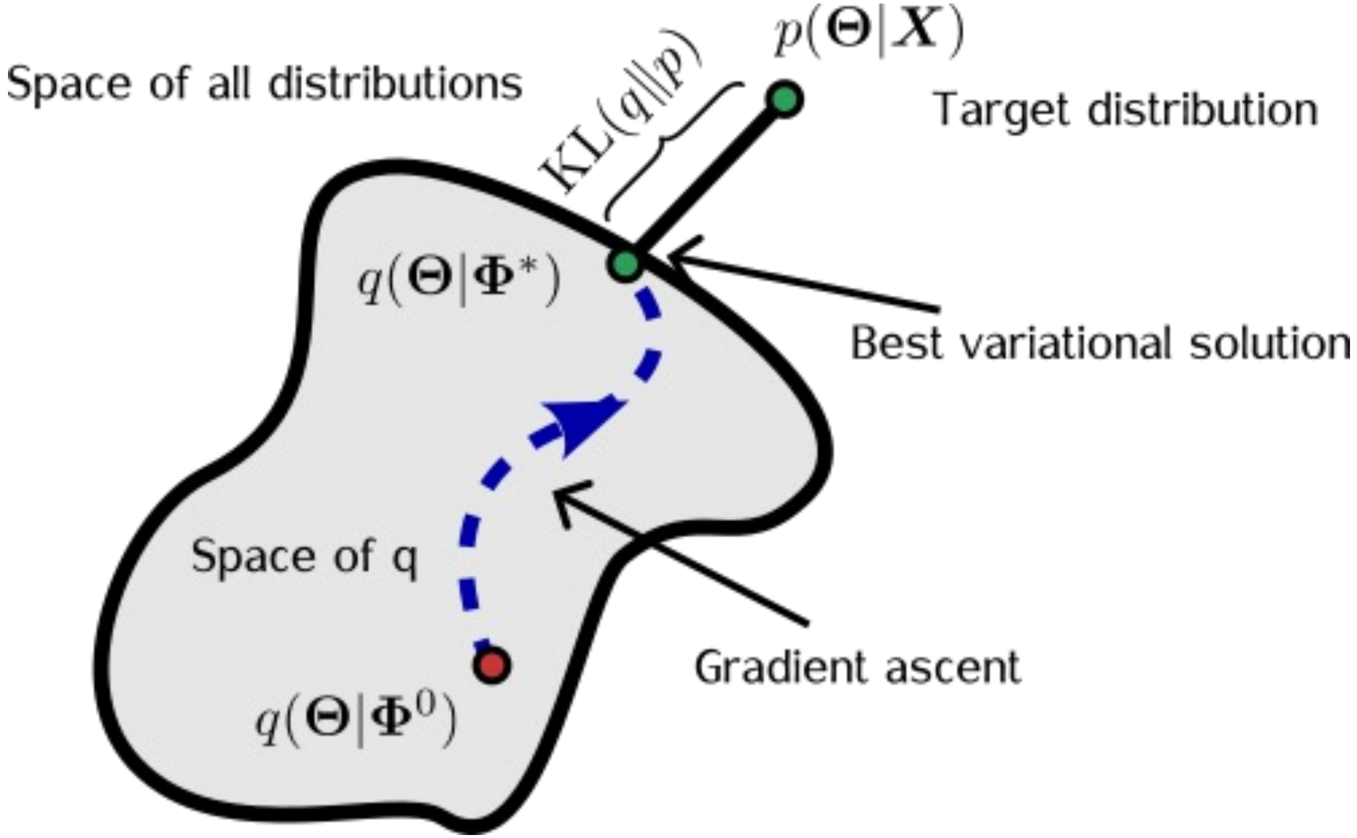
HMC

$$H(\boldsymbol{\theta}, \boldsymbol{r}) = U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{r}^T \boldsymbol{M}^{-1}\boldsymbol{r}$$



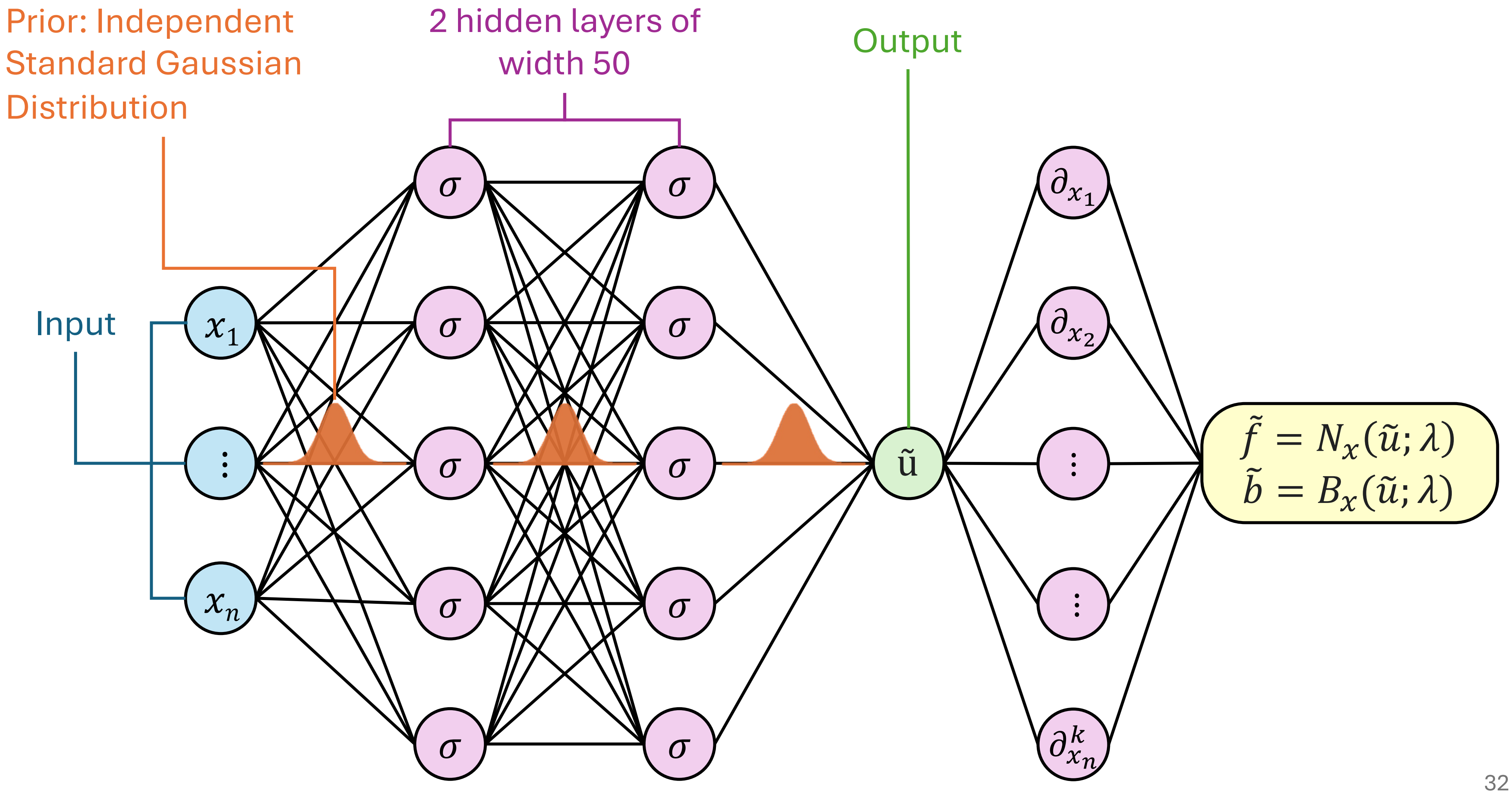
Source: *Masfara, L. O. M., & Weemstra, C. (2024). Hamiltonian Monte Carlo to characterize induced earthquakes: Application to a ML 3.4 event in the Groningen gas field and the role of prior. Earth and Space Science, 11, e2023EA003184. https://doi.org/10.1029/2023EA003184*

VI - Variational Inference



Results

The Network



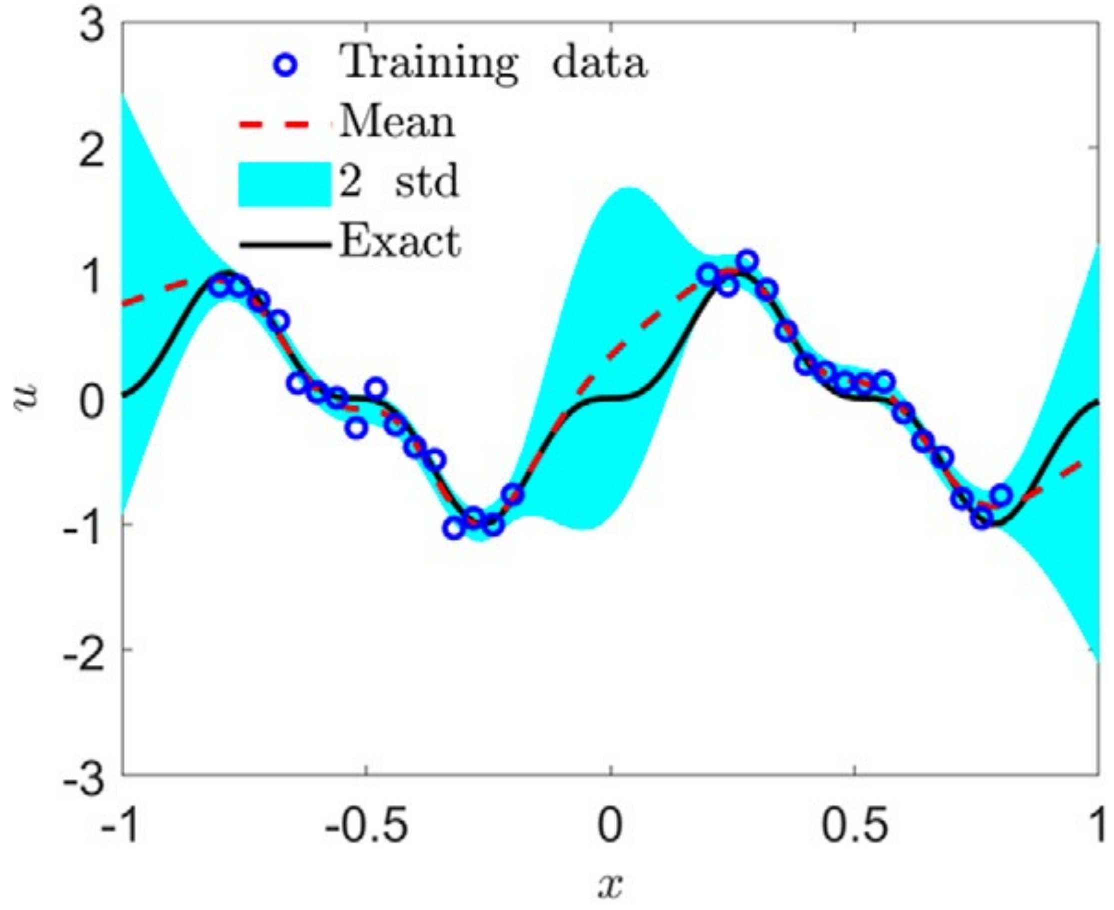
Prior: Independent Standard Gaussian Distribution
2 hidden layers of width 50
Output
Input
x_1
$\vdots$
x_n
σ
ũ
∂_{x_1}
∂_{x_2}
$\vdots$
$\partial^k_{x_n}$
$\tilde{f} = N_x(\tilde{u}; \lambda)$
$\tilde{b} = B_x(\tilde{u}; \lambda)$

Function regression

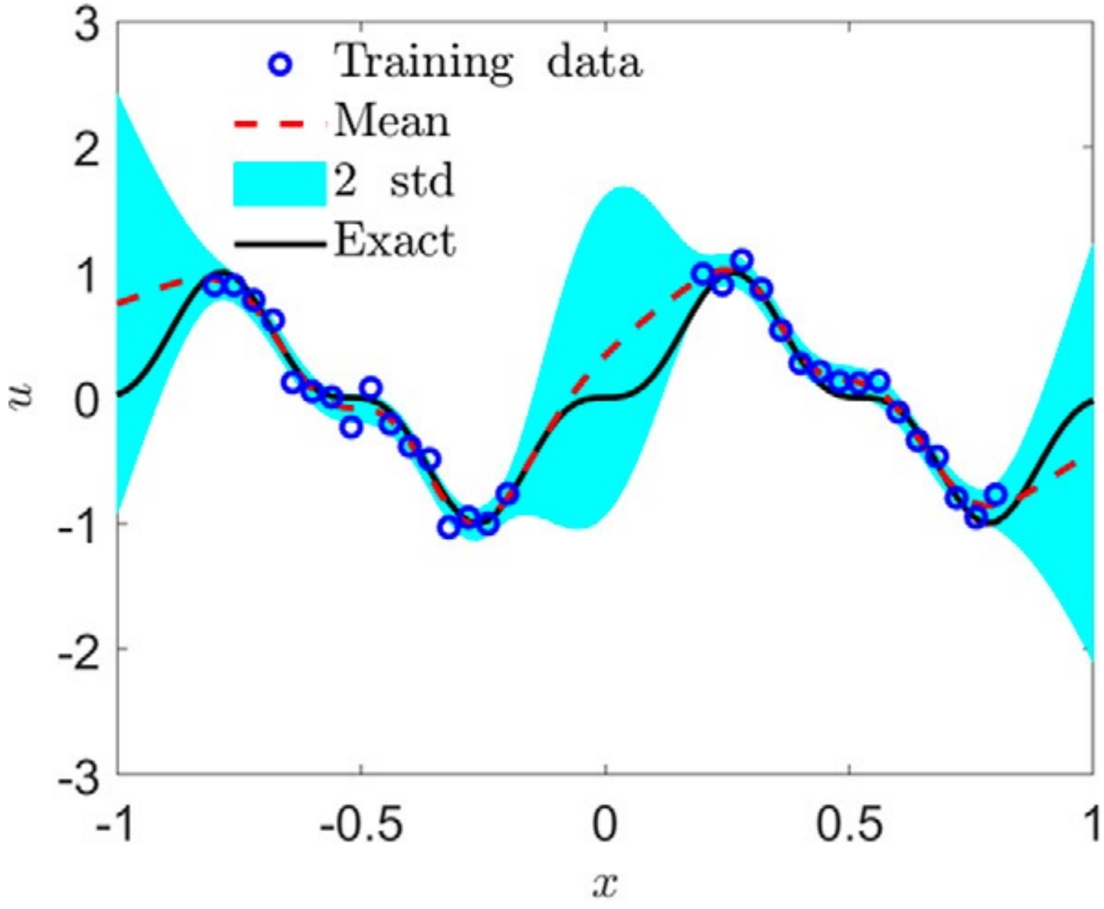
$$u(x) = sin^3(6x), \qquad x \in [-1,1]$$

- BNN instead of B-PINN
- 32 training points $\bar{u}^{(i)}$ in $[-0.8, -0.2] \cup [0.2, 0.8]$

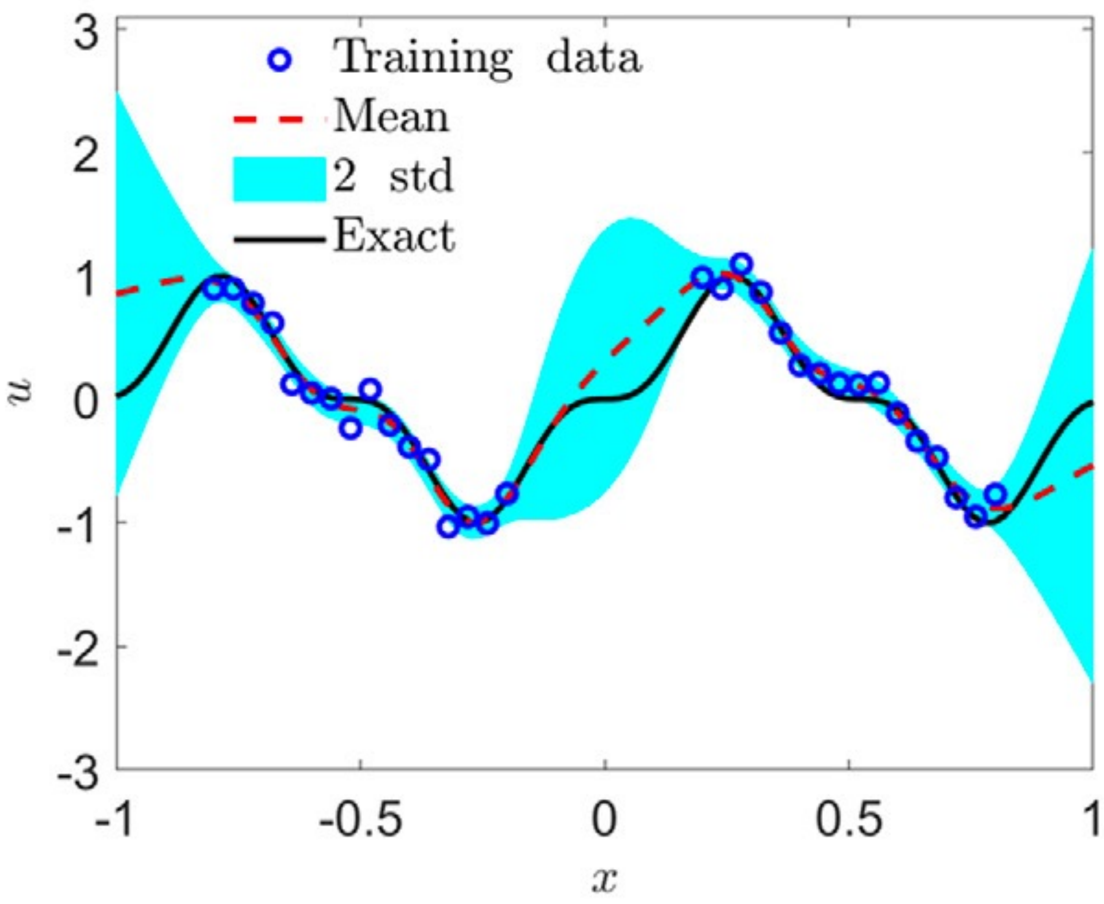
BNN-GPR



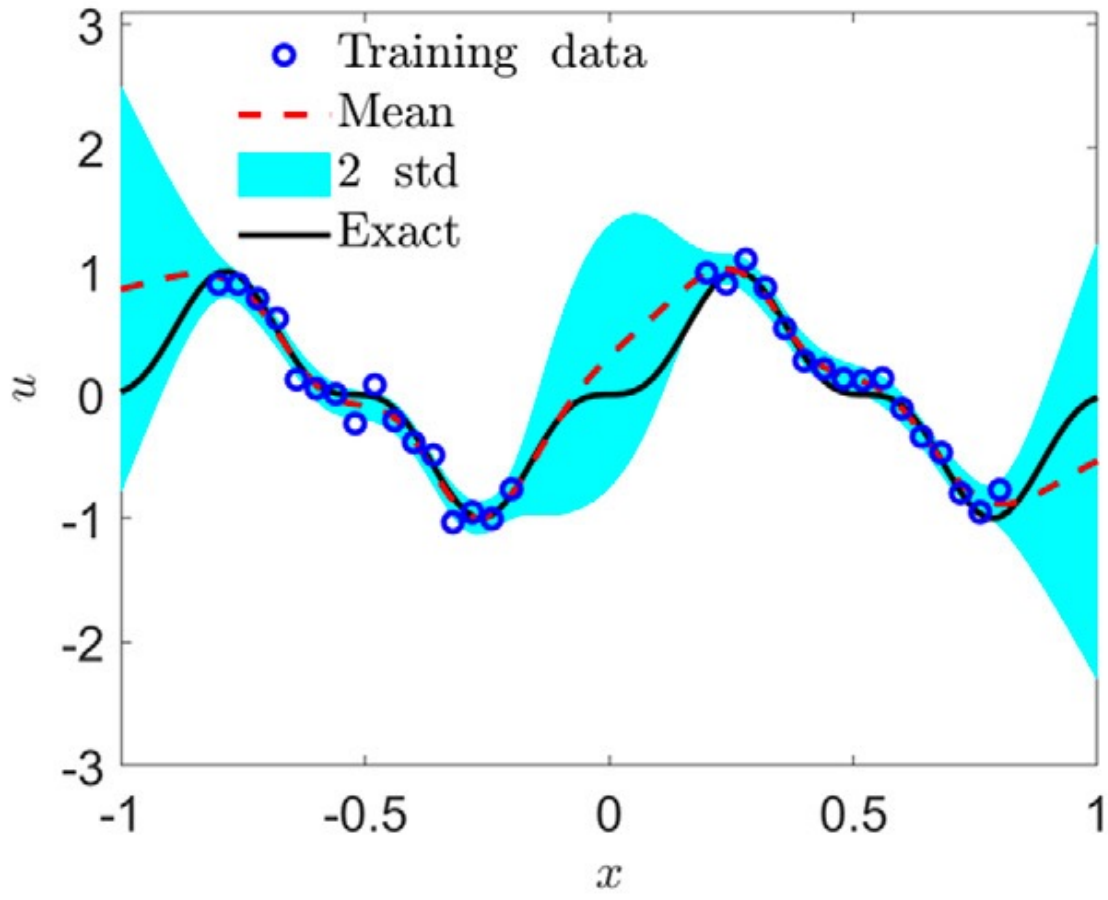
BNN-GPR



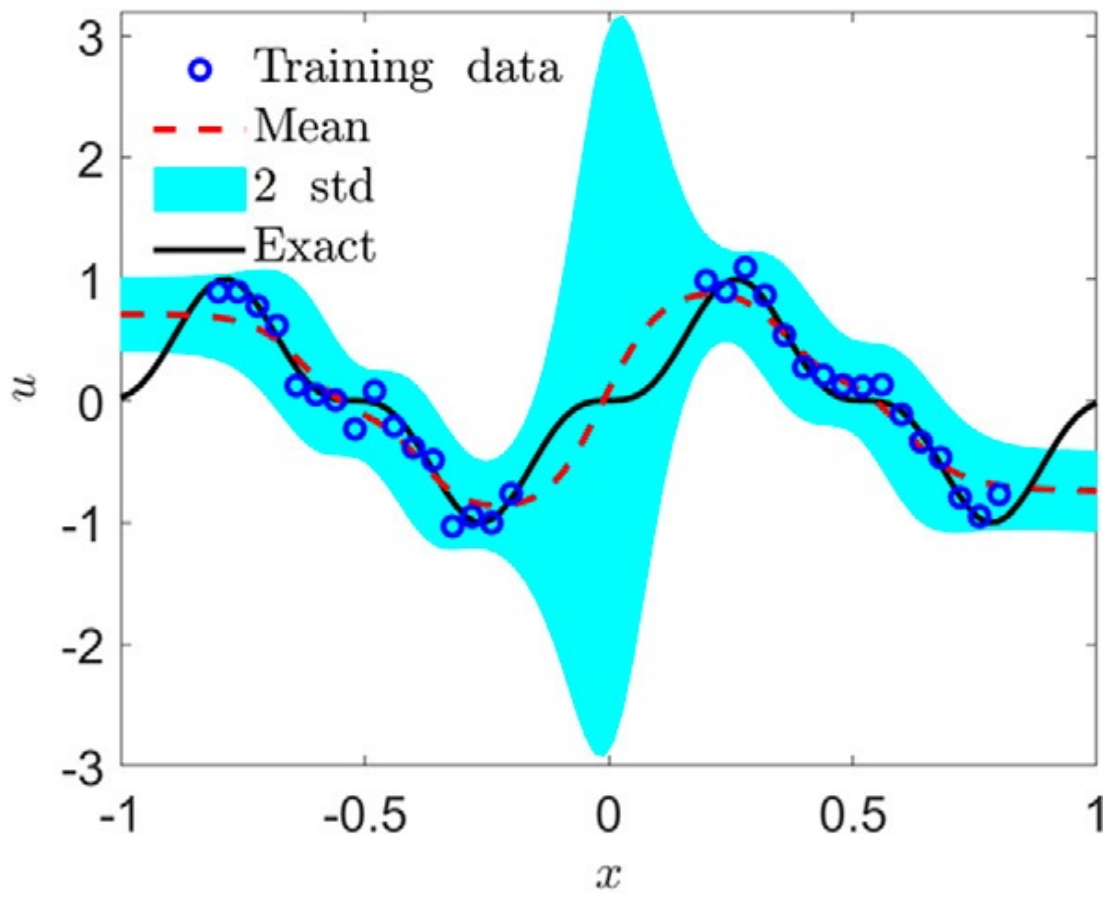
BNN-HMC



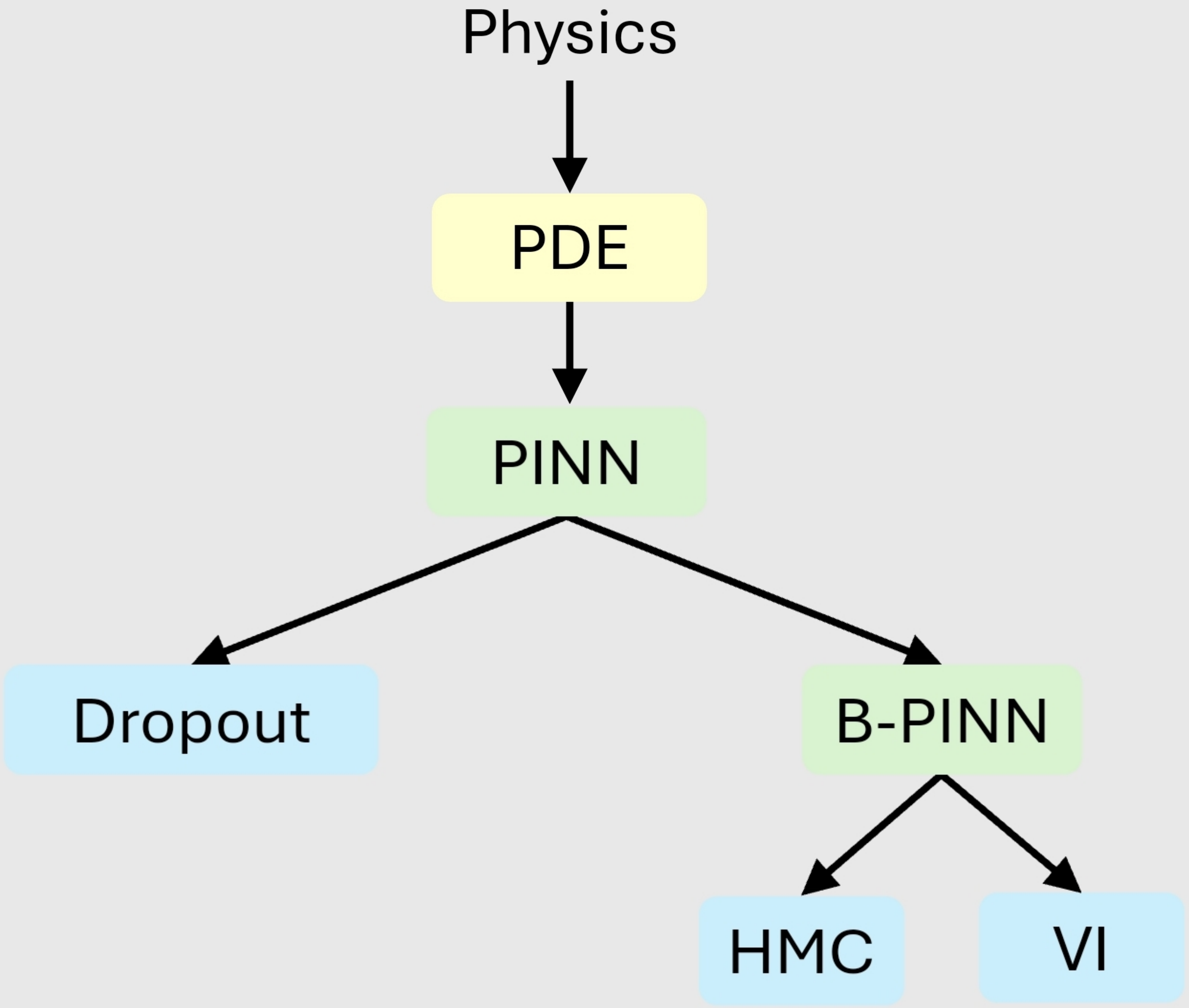
BNN-HMC



BNN-VI



From here on, B-PINNs instead of BNNs are used


Physics
PDE
PINN
Dropout
B-PINN
HMC
VI

1D linear Poisson Equation

$$\mathcal{N}_x(u;\lambda) = f, \qquad x \in D$$

$$\mathcal{B}_x(u;\lambda) = b, \qquad x \in \Gamma$$

$$\lambda \partial_x^2 u = f \qquad x \in [-0.7, 0.7]$$

$$u = b \qquad x \in \{-0.7, 0.7\}$$

1D linear Poisson Equation

differential operator

potential field

charge density

constant factor

$$\lambda \partial_x^2 u = f \qquad x \in [-0.7, 0.7]$$

$$u = b \qquad x \in \{-0.7, 0.7\}$$

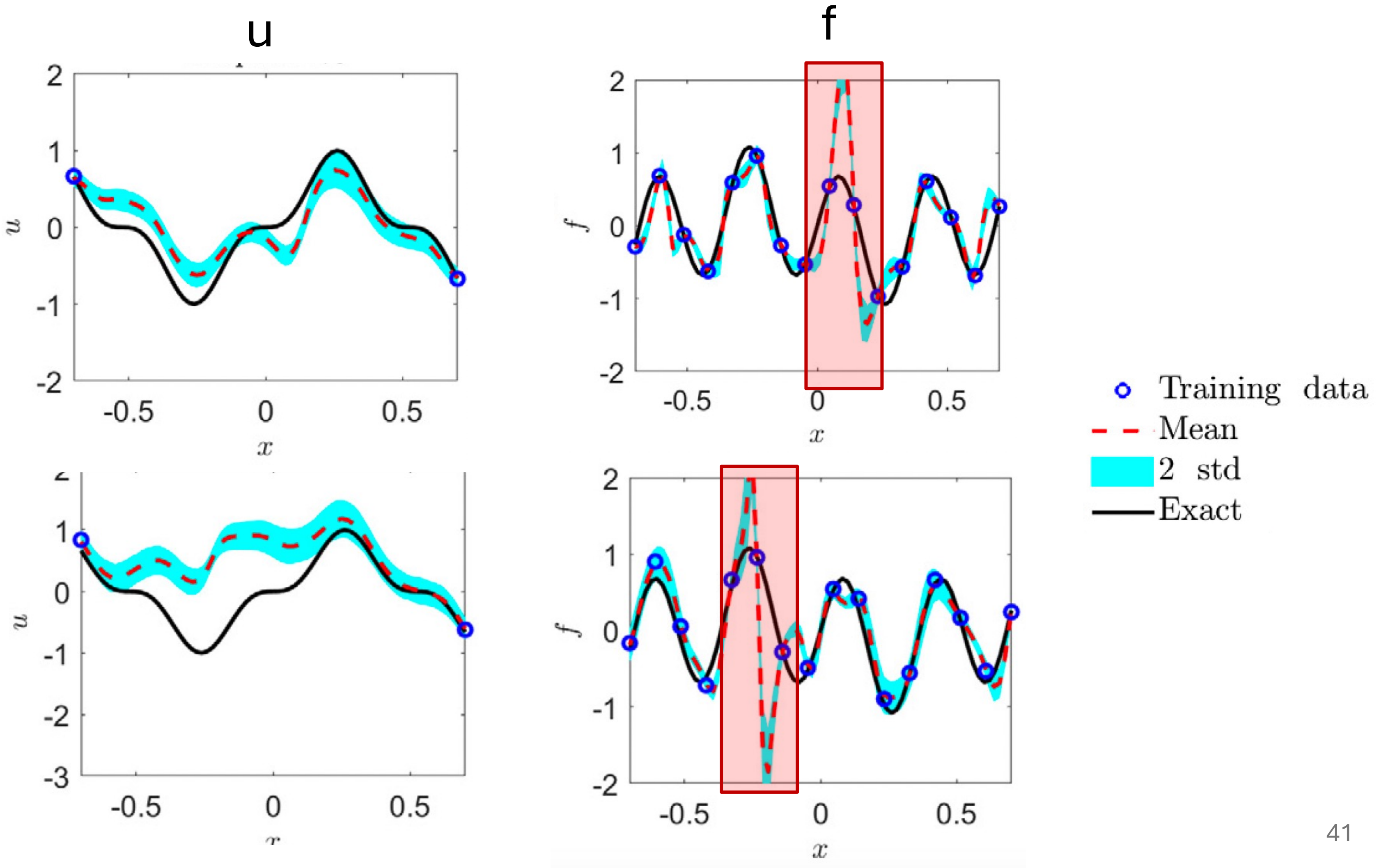
$$u(x) = sin^3(6x)$$

PINN-Dropout 1%

$\lambda \partial_x^2 u = f$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.01^2)$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.1^2)$

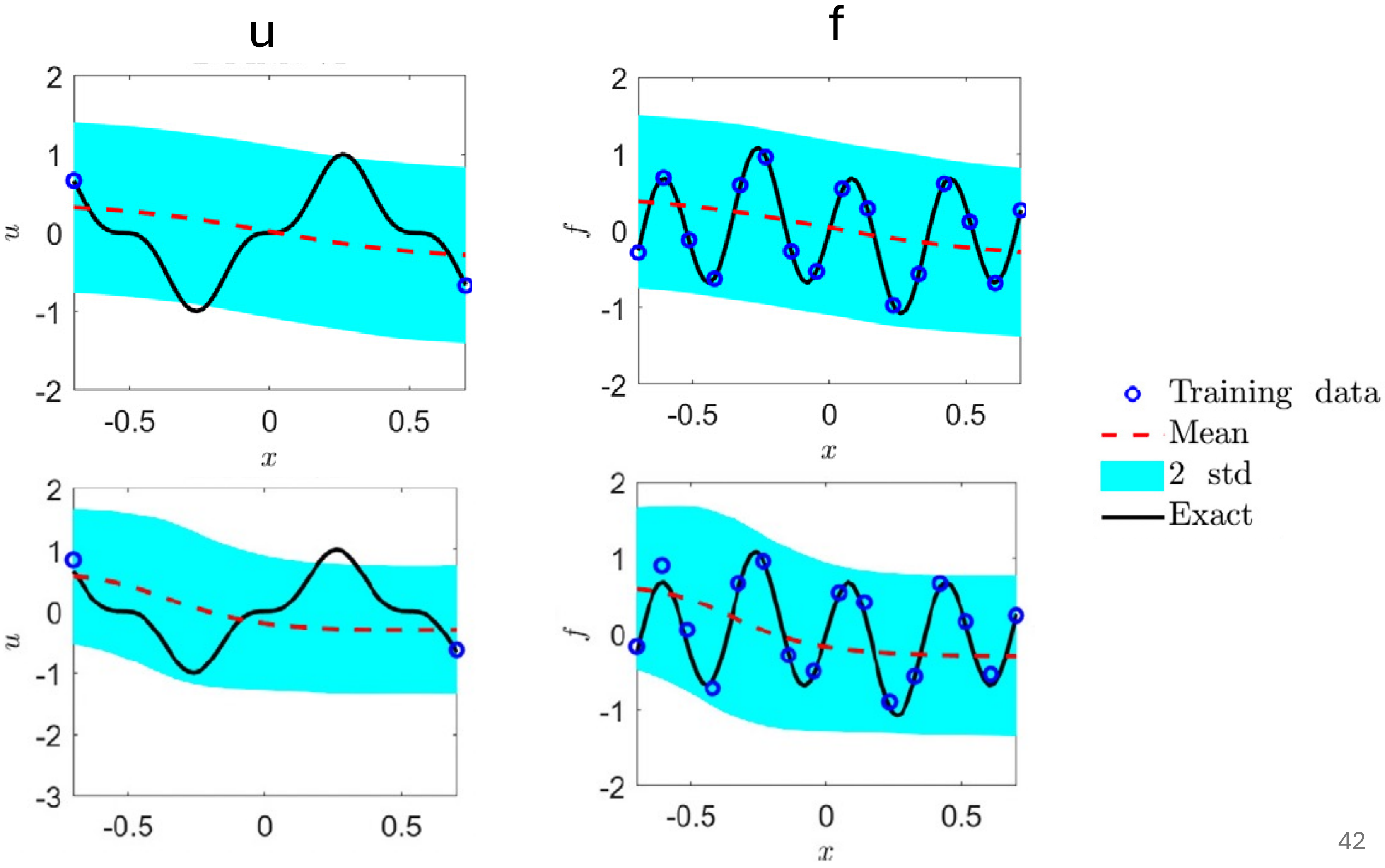


B-PINN-VI

$$\lambda \partial_x^2 u = f$$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.01^2)$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.1^2)$

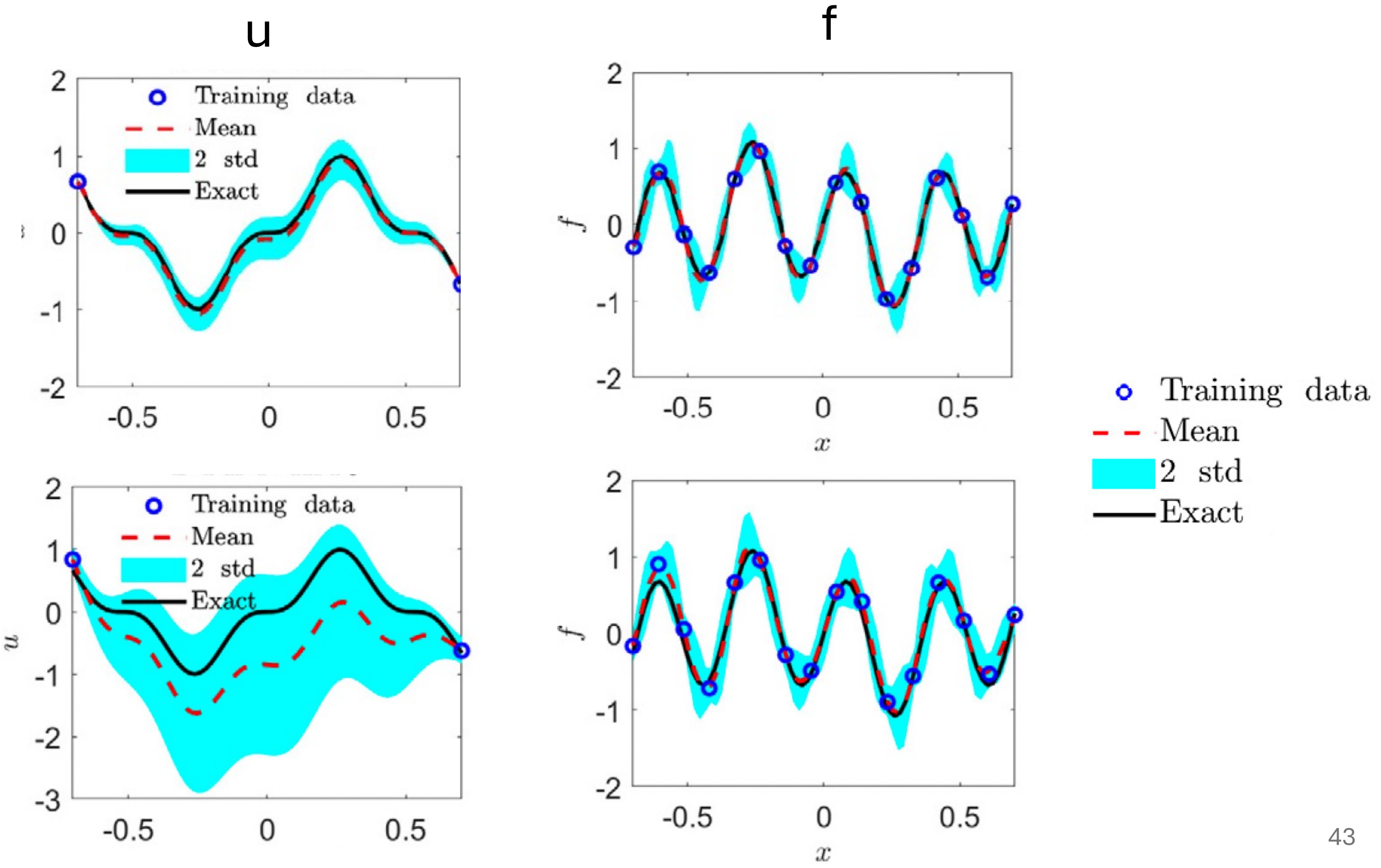


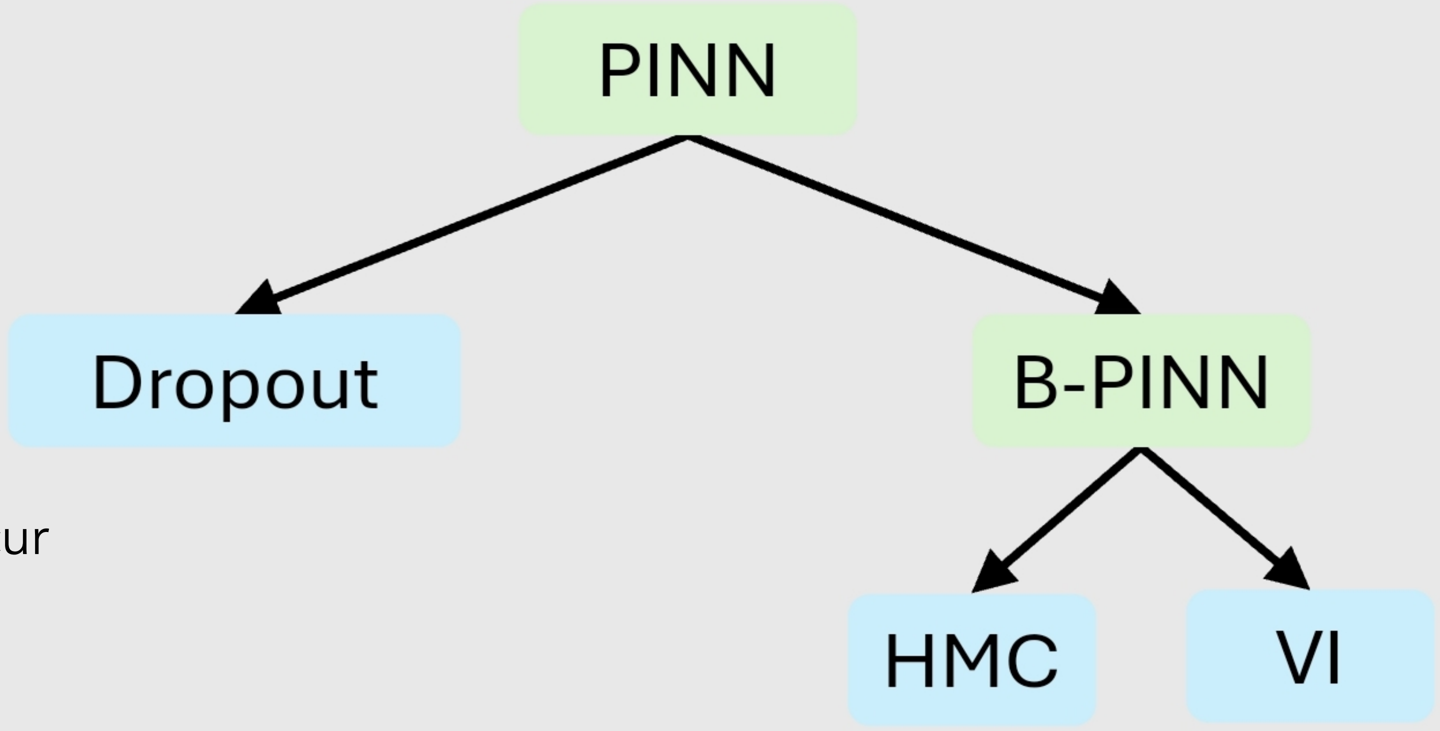
B-PINN-HMC

$$\lambda \partial_x^2 u = f$$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.01^2)$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.1^2)$





- overfitting can occur

- accurate and robust, even for noisy data

- failing can occur

1D nonlinear Poisson Equation

$$\lambda \partial_x^2 u = f \qquad x \in [-0.7, 0.7]$$

$$u = b \qquad x \in \{-0.7, 0.7\}$$

$$\lambda \partial_x^2 u + k \tanh(u) = f \qquad x \in [-0.7, 0.7]$$

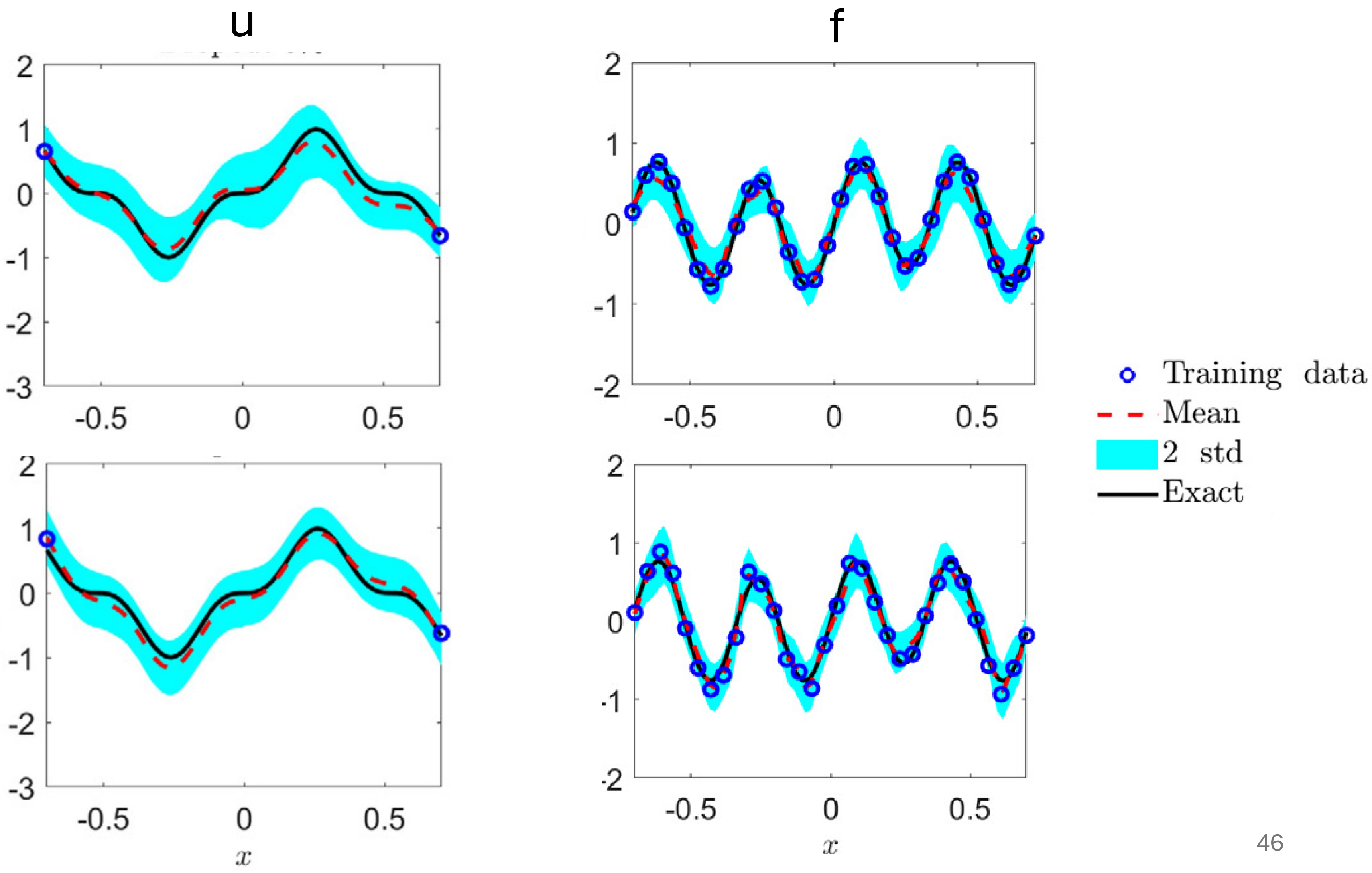
$$u = b \qquad x \in \{-0.7, 0.7\}$$

$$\lambda \partial_x^2 u + k \tanh(u) = f$$

PINN-Dropout 5%

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.01^2)$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.1^2)$

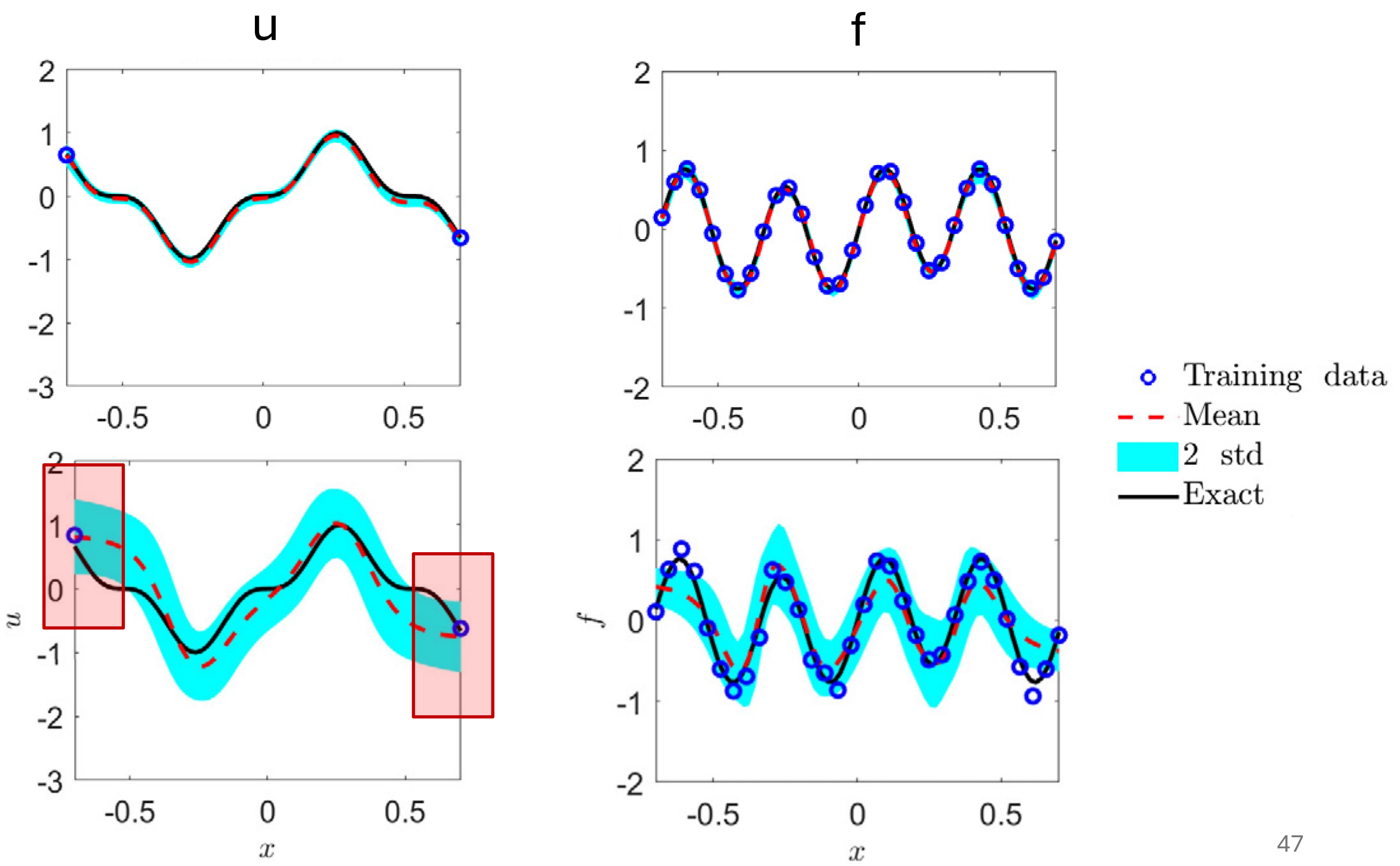


B-PINN-VI

$$\lambda \partial_x^2 u + k \tanh(u) = f$$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.01^2)$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.1^2)$

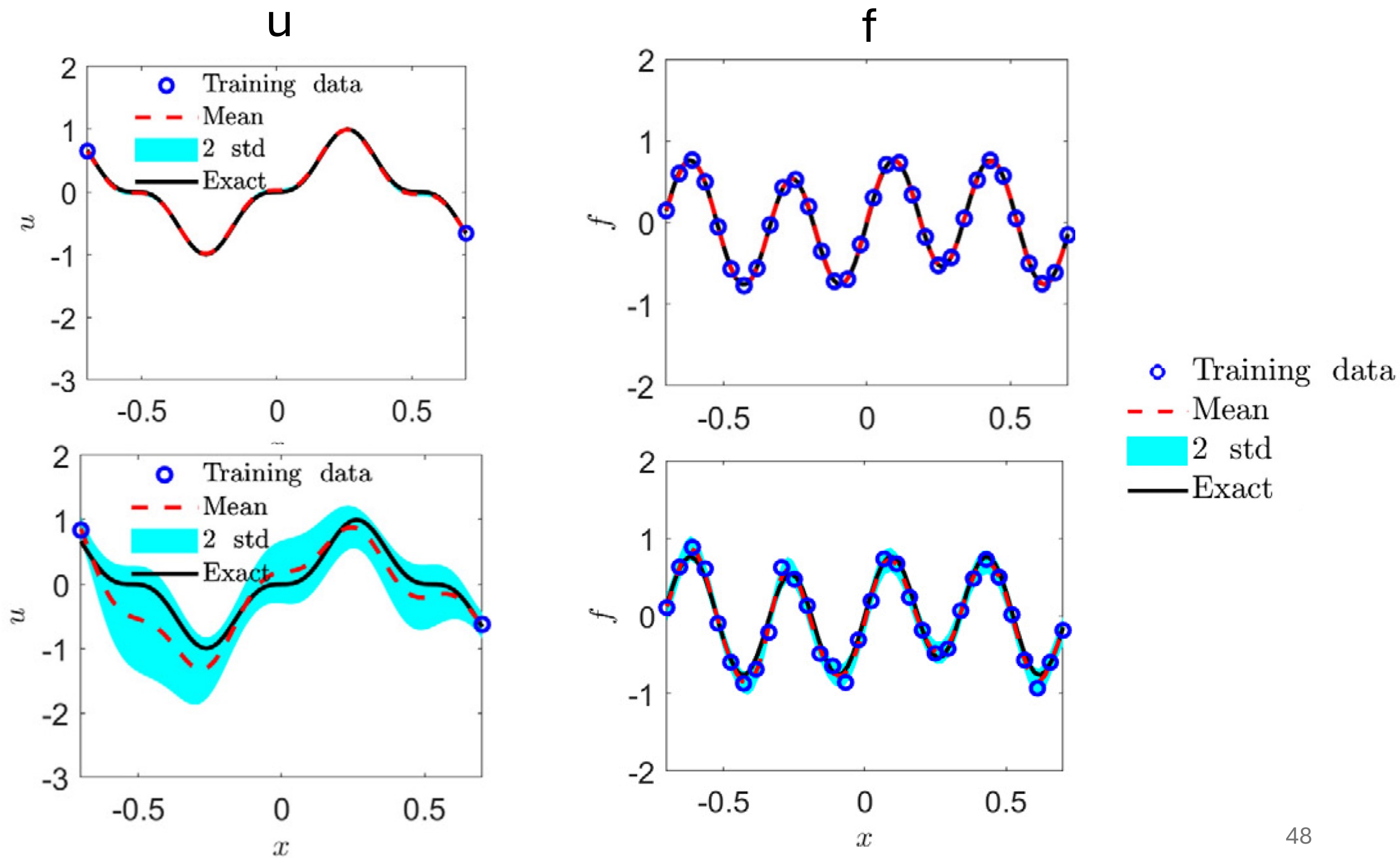


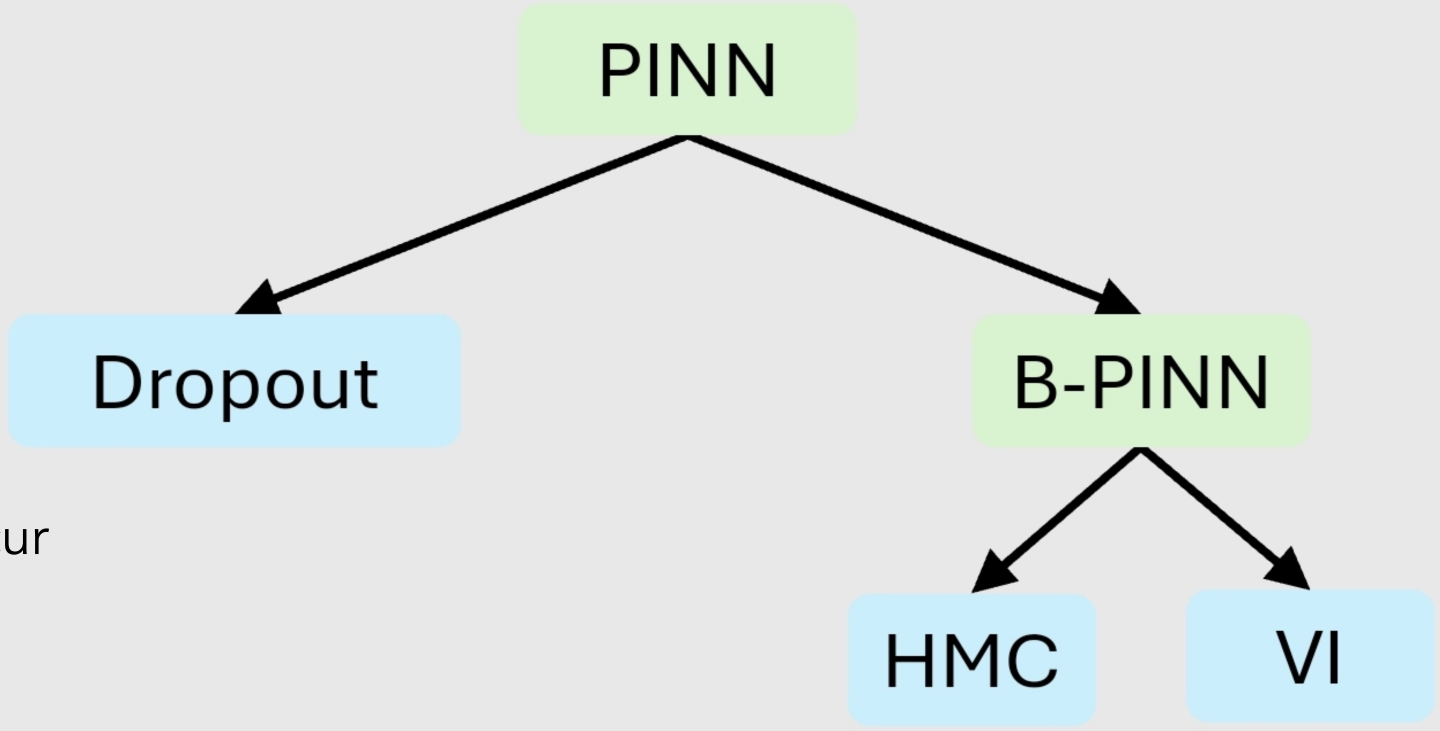
B-PINN-HMC

$$\lambda \partial_x^2 u + k \tanh(u) = f$$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.01^2)$

$\epsilon_f, \epsilon_b \in \mathcal{N}(0, 0.1^2)$

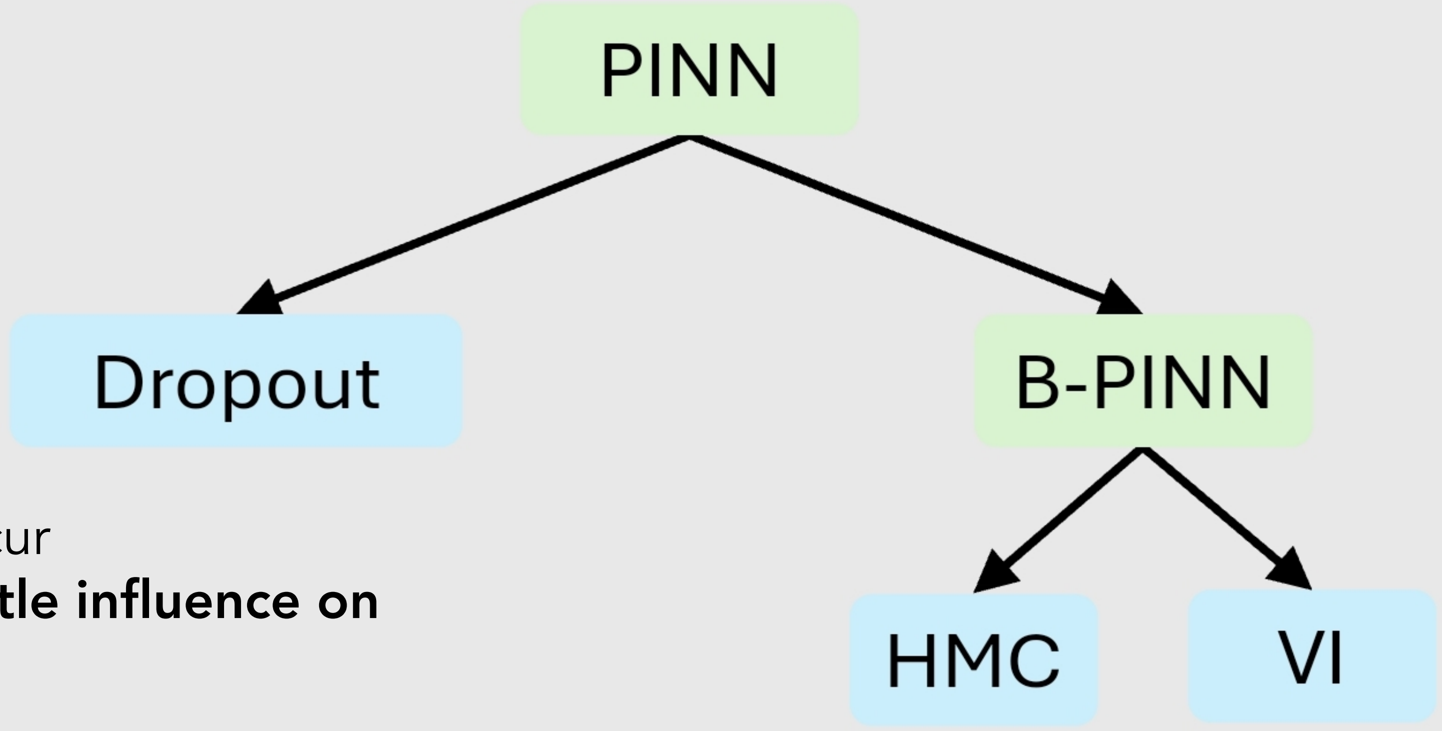




- overfitting can occur

- accurate and robust, even for noisy data

- failing can occur



- overfitting can occur
- **noise can have little influence on (un-)certainty**

- accurate and robust, even for noisy data

- failing can occur
- **unreasonable uncertainty at the boundaries**

1D nonlinear Poisson Equation – inverse problem

$$\lambda \partial_x^2 u + k \tanh(u) = f \qquad x \in [-0.7, 0.7]$$

$$u = b \qquad x \in \{-0.7, 0.7\}$$

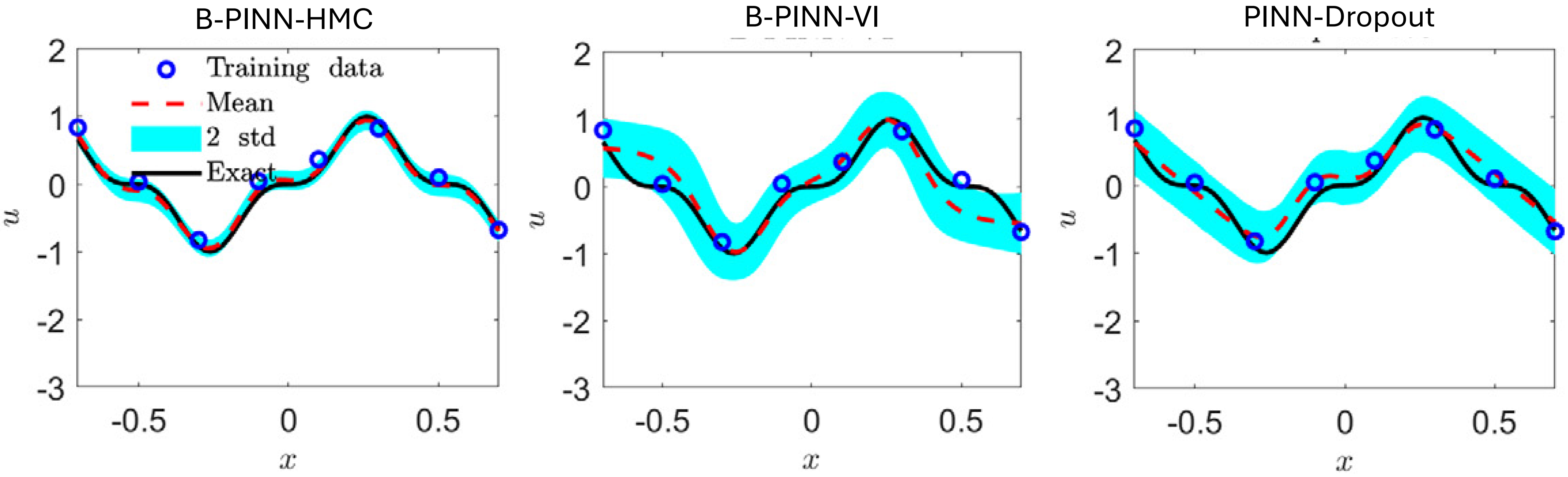
↓

$$\lambda \partial_x^2 u + k \tanh(u) = f \qquad x \in [-0.7, 0.7]$$

$$u = b \qquad x \in \{-0.7, 0.7\}$$

1D nonlinear Poisson Equation – inverse problem

$$\lambda \partial_x^2 u + k \tanh(u) = f$$



$$\epsilon_f \sim \mathcal{N}(0, 0.1^2),\ \epsilon_u \sim \mathcal{N}(0, 0.1^2),\ \epsilon_b \sim \mathcal{N}(0, 0.1^2)$$

1D nonlinear Poisson Equation – inverse problem

$$\lambda \partial_x^2 u + k \tanh(u) = f$$

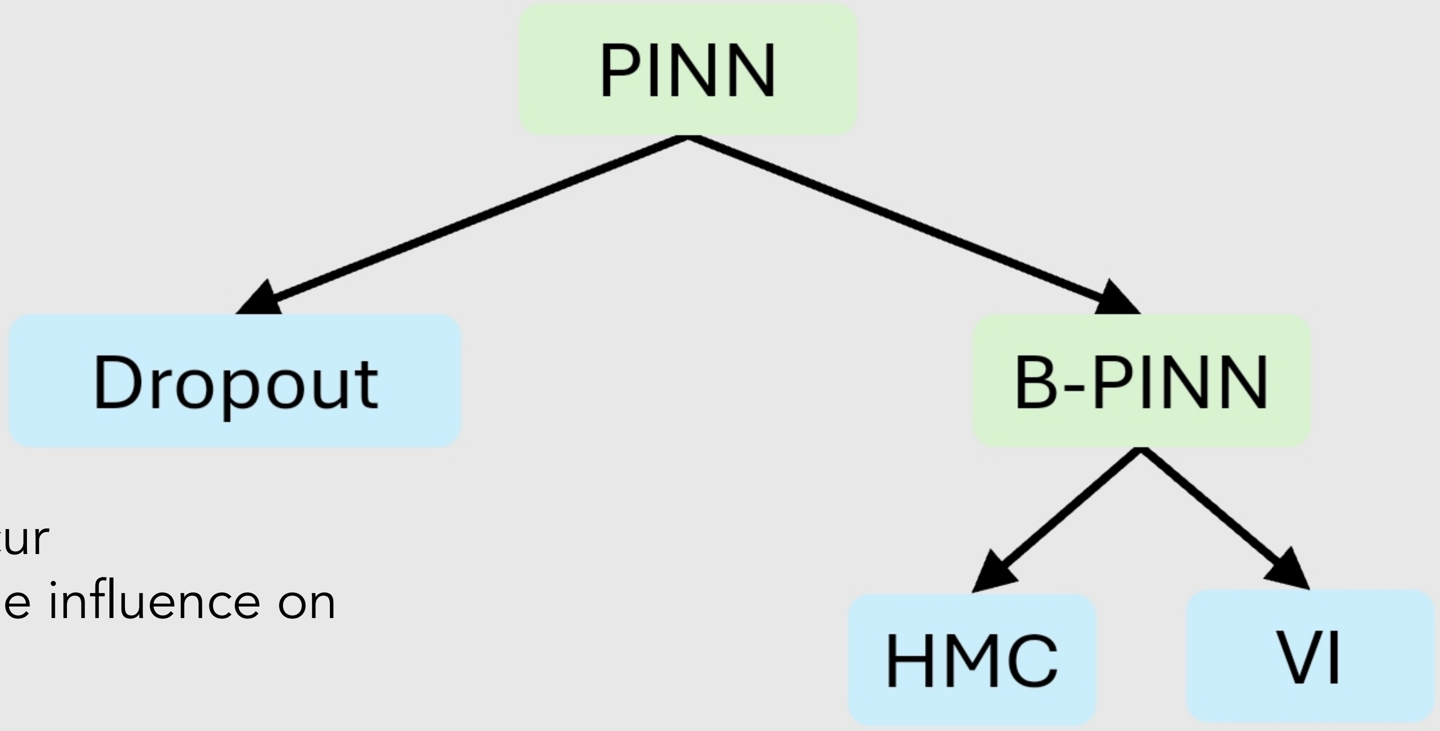
Exact value for k is 0.7

Noise scale		B-PINN-HMC	B-PINN-VI	Dropout-1%	Dropout-5%
0.01	Mean	0.705	0.708	0.714	0.669
	Std	5.75×10^{-3}	4.01×10^{-3}	4.38×10^{-3}	2.02×10^{-2}
0.1	Mean	0.665	0.775	0.746	0.633
	Std	5.63×10^{-2}	3.58×10^{-2}	6.508×10^{-3}	6.45×10^{-3}

B-PINN-HMC: quite accurate, reasonable uncertainties

B-PINN-VI: higher error than HMC, reasonable uncertainties

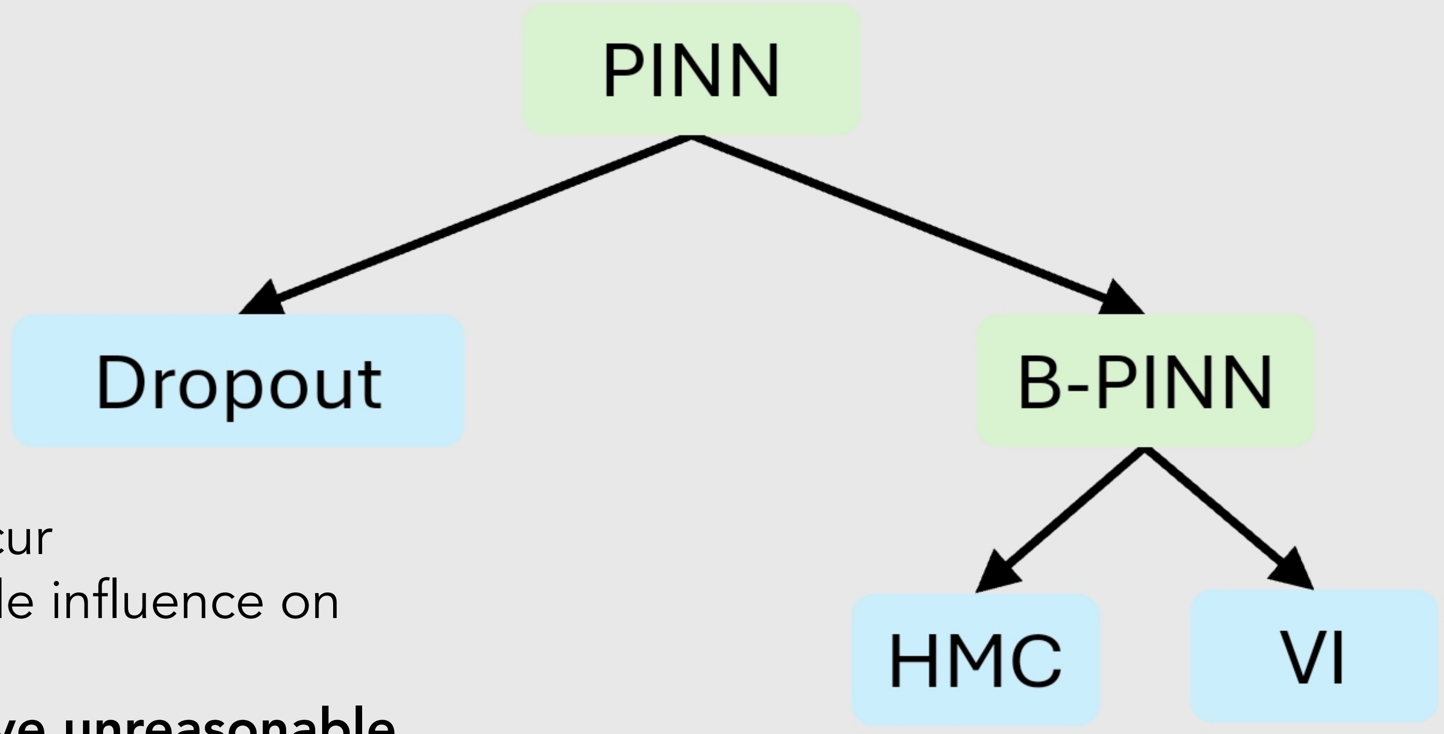
Dropout: higher error than HMC and VI, unresonable uncertanties



- overfitting can occur
- noise can have little influence on (un-)certainty

- accurate and robust, even for noisy data

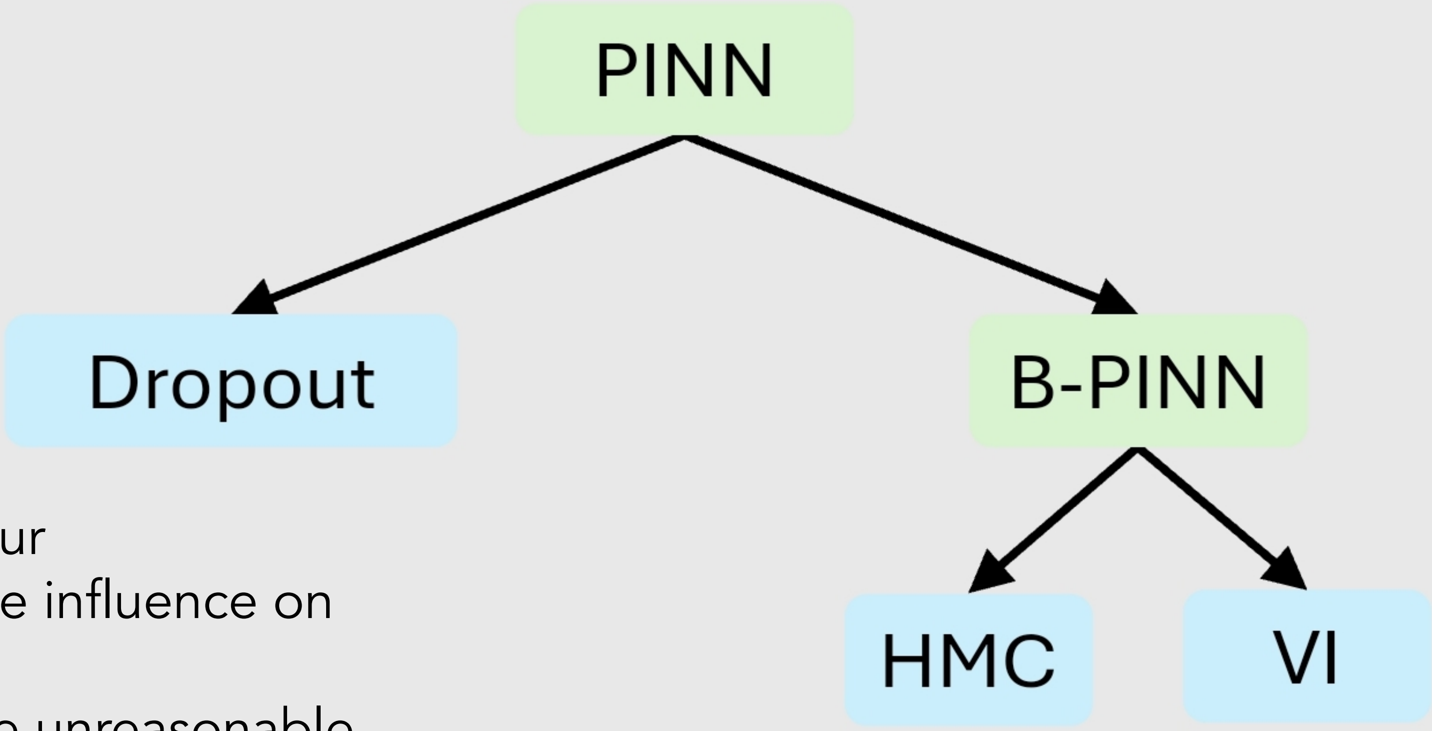
- failing can occur
- unreasonable uncertainty at the boundaries



- overfitting can occur
- noise can have little influence on (un-)certainty
- **noise can also have unreasonable influence on (un-)certainty of the model**

- accurate and robust, even for noisy data

- failing can occur
- unreasonable uncertainty at the boundaries
- **higher error than HMC**



Dropout:

- overfitting can occur
- noise can have little influence on (un-)certainty
- noise can also have unreasonable influence on (un-)certainty

HMC:

- accurate and robust, even for noisy data

VI:

- failing can occur
- unreasonable uncertainty at the boundaries
- higher error than HMC

Open problems:
- Choice of e.g. prior distribution
- Big data case

EFFICIENT BAYESIAN PHYSICS INFORMED NEURAL NETWORKS FOR INVERSE PROBLEMS VIA ENSEMBLE KALMAN INVERSION

ANDREW PENSONEAULT* AND XUEYU ZHU†

Abstract. Bayesian Physics Informed Neural Networks (B-PINNs) have gained significant attention for inferring physical parameters and learning the forward solutions for problems based on partial differential equations. However, the overparameterized nature of neural networks poses a computational challenge for high-dimensional posterior inference. Existing inference approaches, such as particle-based or variance inference methods, are either computationally expensive for high-dimensional posterior inference or provide unsatisfactory uncertainty estimates. In this paper, we present a new efficient inference algorithm for B-PINNs that uses Ensemble Kalman Inversion (EKI) for high-dimensional inference tasks. By reframing the setup of B-PINNs as a traditional Bayesian inverse problem, we can take advantage of EKI's key features: (1) gradient-free, (2) computational complexity scales linearly with the dimension of the parameter spaces, and (3) rapid convergence with typically $\mathcal{O}(100)$ iterations. We demonstrate the applicability and performance of the proposed method through various types of numerical examples. We find that our proposed method can achieve inference results with informative uncertainty estimates comparable to Hamiltonian Monte Carlo (HMC)-based B-PINNs with a much reduced computational cost. These findings suggest that our proposed approach has great potential for uncertainty quantification in physics-informed machine learning for practical applications.

2.2. Hamiltonian Monte Carlo (HMC). Next, we briefly review Hamiltonian Monte Carlo (HMC), a popular inference algorithm for B-PINNs [48] that serves as a baseline method for our proposed method. Hamiltonian Monte Carlo (HMC)

[48] Liu Yang, Xuhui Meng, and George Em Karniadakis. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425:109913, jan 2021.

https://www.sciencedirect.com/science/article/abs/pii/S0021999124002559