B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data

Liu Yang , Xuhui Meng , George Em Karniadakis

Journal of Computational Physics

Volume 425, 15 January 2021, 109913

- In practical applications, the measurement of data is usually noisy (produced by the sensors).
- There is no quantification of uncertainty in the PINN framework.

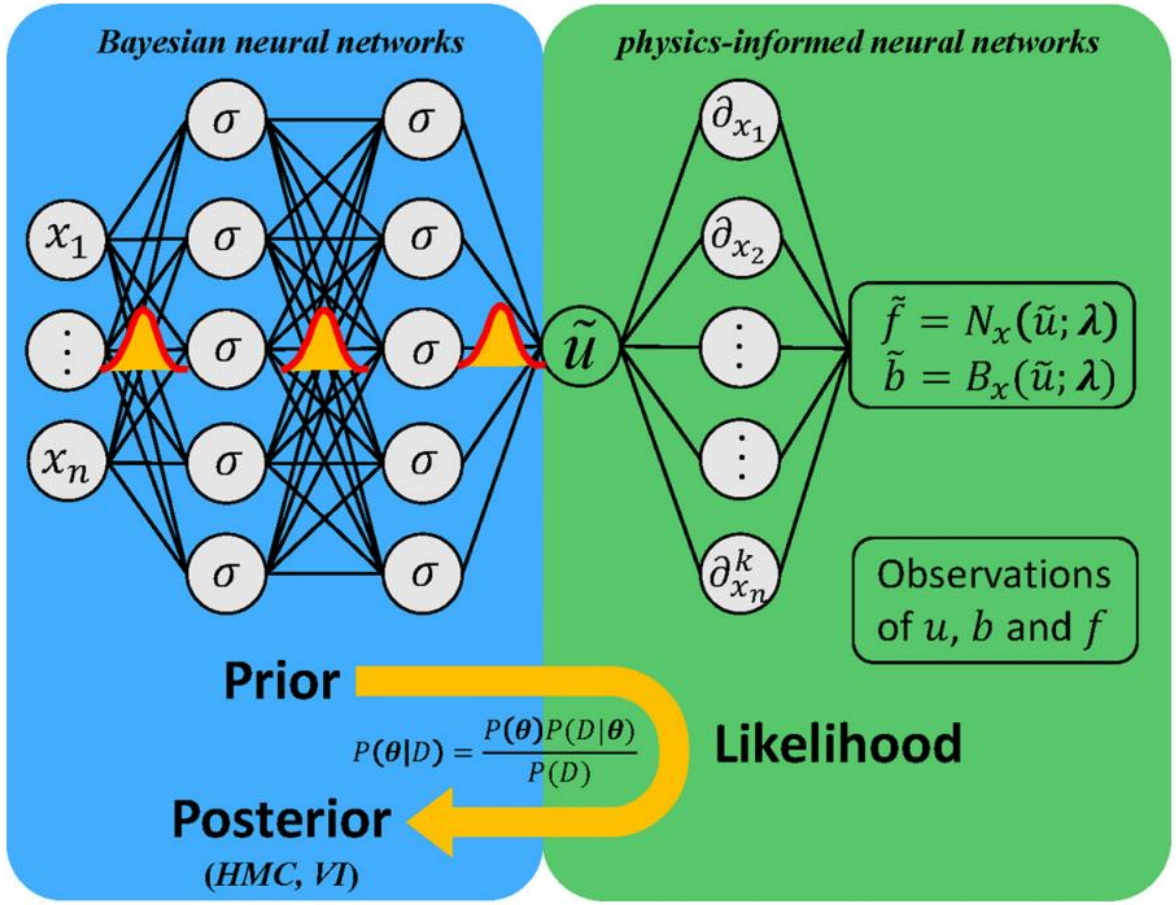


Fig. 1. Schematic of the Bayesian physics-informed neural network (B-PINN). $P(\boldsymbol{\theta})$ is the prior for hyperparameters as well as the unknown terms in PDEs, $P(\mathcal{D}|\boldsymbol{\theta})$ represents the likelihood of observations (e.g., u, b, f), and $P(\boldsymbol{\theta}|\mathcal{D})$ is the posterior. The blue panel represents the Bayesian neural network while the green panel represents the physics-informed part. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

- a general partial differential equation (PDE) of the form

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

- Available dataset D are scattered noisy measurements of u, f and b from sensors:

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

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

- Bayesian framework starts from representing u with a surrogate model u˜(x; θ):

$$\tilde{f}(\boldsymbol{x};\boldsymbol{\theta}) = \mathcal{N}_{\boldsymbol{x}}(\tilde{u}(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{\lambda}), \ \tilde{b}(\boldsymbol{x};\boldsymbol{\theta}) = \mathcal{B}_{\boldsymbol{x}}(\tilde{u}(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{\lambda}).$$

- the likelihood can be calculated as:

$$P(\mathcal{D}|\boldsymbol{\theta}) = P(\mathcal{D}_u|\boldsymbol{\theta})P(\mathcal{D}_f|\boldsymbol{\theta})P(\mathcal{D}_b|\boldsymbol{\theta}),$$

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

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

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

- the posterior is obtained from Bayes' theorem:

$$P(\boldsymbol{\theta}|\mathcal{D}) = \frac{P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathcal{D})} \simeq P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta}),$$

- In the case of inverse problems, we also need to assign a prior distribution for λ, which could be independent of P(θ).

$$P(\boldsymbol{\theta}, \boldsymbol{\lambda}|\mathcal{D}) = \frac{P(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\lambda})P(\boldsymbol{\theta}, \boldsymbol{\lambda})}{P(\mathcal{D})} \simeq P(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\lambda})P(\boldsymbol{\theta}, \boldsymbol{\lambda}) = P(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\lambda})P(\boldsymbol{\theta})P(\boldsymbol{\lambda}),$$

- In the application of Bayesian neural networks, a commonly used prior for θ is that each component of θ is an independent Gaussian distribution.
- In this case, the prior of the function u˜(x) is actually a Gaussian process as the width of hidden layers goes to infinity with √ Nlσw,l

$$\boldsymbol{z}_l = \phi(\boldsymbol{w}_{l-1}\boldsymbol{z}_{l-1} + \boldsymbol{b}_{l-1}), \quad l = 1, 2...L,$$
$$\tilde{u} = \boldsymbol{w}_L\boldsymbol{z}_L + \boldsymbol{b}_L,$$

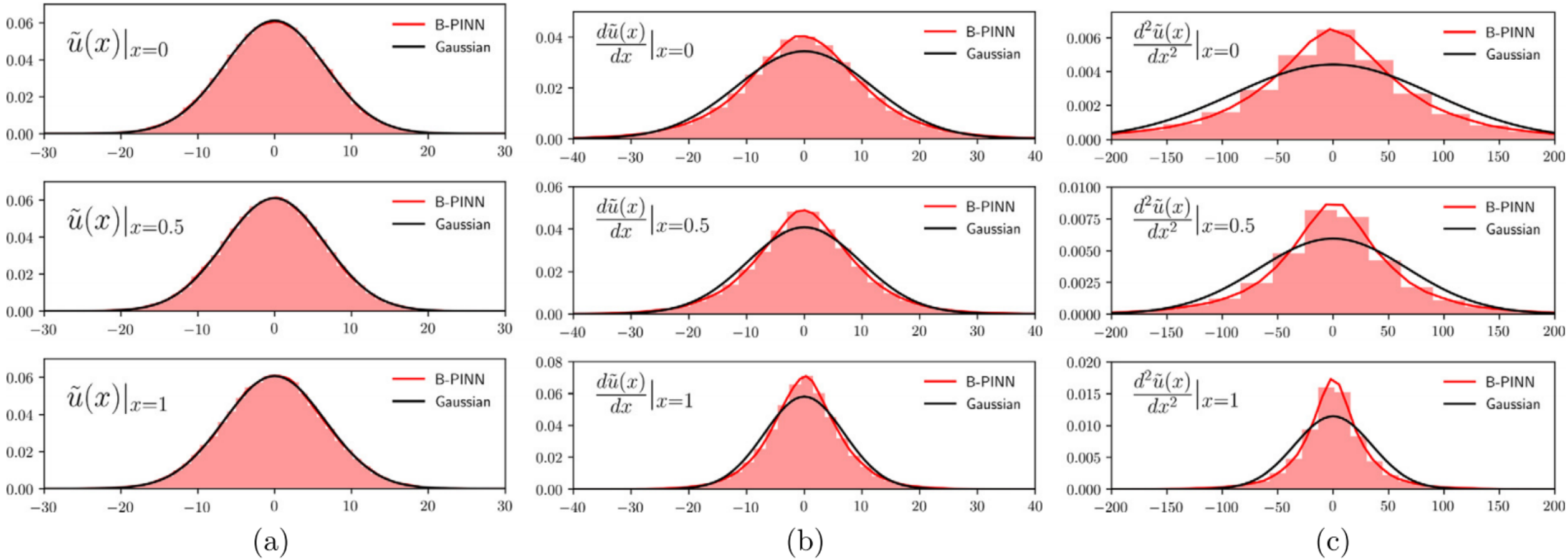


Fig. 2. Comparison between the B-PINNs' prior distributions and Gaussian distributions. The red lines and histograms represent the density of B-PINNs' outputs (a) $\tilde{u}(x)$, (b) $d\tilde{u}(x)/dx$, and (c) $d^2\tilde{u}(x)/dx^2$ at $x = 0$, 0.5, and 1. The black lines are the density functions of the corresponding Gaussian distributions with zero mean and the same standard deviations as the B-PINNs' outputs.

Hamiltonian Monte Carlo (HMC) method

- an efficient Markov Chain Monte Carlo (MCMC) method based on the Hamiltonian dynamics

$$p(\boldsymbol{x}) \propto \exp\left(-\frac{U(\boldsymbol{x})}{T}\right)$$

- P(x) represents the probability that a system is in a state x.
- P(x) depends numerically on the state's energy U(x) and the corresponding temperature T.

$$p(\boldsymbol{x}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- the target posterior distribution for θ given a certain number of observations D is defined as

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

$$\downarrow$$

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

$$\downarrow$$

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

- HMC generates samples from a joint distribution of (θ, r) as follows. We simply discard the r samples, the θ samples have marginal distribution P(θ |D).

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

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.

the samples are generated from the following Hamiltonian dynamics

$$d\boldsymbol{\theta} = \boldsymbol{M}^{-1} \boldsymbol{r} dt,$$

$$d\boldsymbol{r} = -\nabla U(\boldsymbol{\theta}) dt.$$

Variational inference (VI) method

- P(θ|D) is approximated by another density function Q (θ; ζ) parameterized by ζ .
- Usually use multiple independent Gaussian distributions. Assuming that the parameters obey Gaussian distribution with mean ζμ,i and standard deviation ln(1 + exp(ζρ,i))

$$Q(\boldsymbol{\theta};\boldsymbol{\zeta}) = \prod_{i=1}^{d_{\boldsymbol{\theta}}} q(\theta_i;\zeta_{\mu,i},\zeta_{\rho,i}),$$

- tune ζ to minimize

$$D_{KL}(Q(\boldsymbol{\theta};\boldsymbol{\zeta})||P(\boldsymbol{\theta}|\mathcal{D})) \simeq \mathbb{E}_{\boldsymbol{\theta}\sim Q}[\ln Q(\boldsymbol{\theta};\boldsymbol{\zeta}) - \ln P(\boldsymbol{\theta}) - \ln P(\mathcal{D}|\boldsymbol{\theta})],$$

Algorithm 2 Variational inference.

Require: an initial state for $\boldsymbol{\zeta}$.

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

Sample $\{\boldsymbol{z}^{(j)}\}_{j=1}^{N_z}$ independently from $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d_{\boldsymbol{\theta}}})$.

$\boldsymbol{\theta}^{(j)} \leftarrow \boldsymbol{\zeta}_{\mu} + \ln(1 + \exp(\boldsymbol{\zeta}_{\rho})) \odot \boldsymbol{z}^{(j)}$, $j = 1, 2 \ldots N_z$, $\odot$ denotes element-wise product.

$L(\boldsymbol{\zeta}) \leftarrow \frac{1}{N_z} \sum_{j=1}^{N_z} [\ln Q(\boldsymbol{\theta}^{(j)}; \boldsymbol{\zeta}) - \ln P(\boldsymbol{\theta}^{(j)}) - \ln P(\mathcal{D}|\boldsymbol{\theta}^{(j)})]$.

Update $\boldsymbol{\zeta}$ with gradient $\nabla_{\boldsymbol{\zeta}} L(\boldsymbol{\zeta})$ using Adam optimizer.

end for

Sample $\{\boldsymbol{z}^{(j)}\}_{j=1}^{M}$ independently from $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d_{\boldsymbol{\theta}}})$.

$\boldsymbol{\theta}^{(j)} \leftarrow \boldsymbol{\zeta}_{\mu} + \ln(1 + \exp(\boldsymbol{\zeta}_{\rho})) \odot \boldsymbol{z}^{(j)}$, $j = 1, 2 \ldots M$.

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

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

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

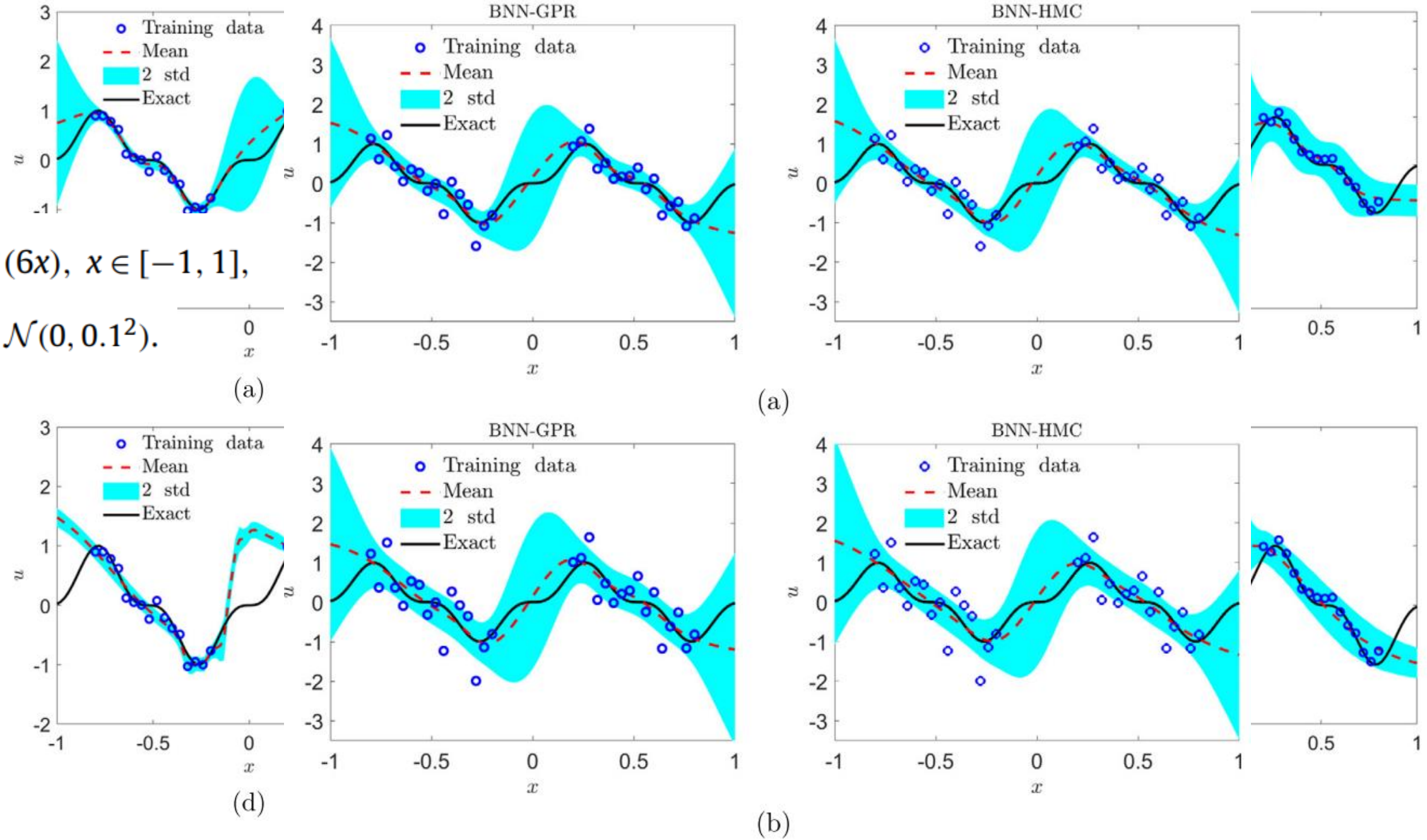
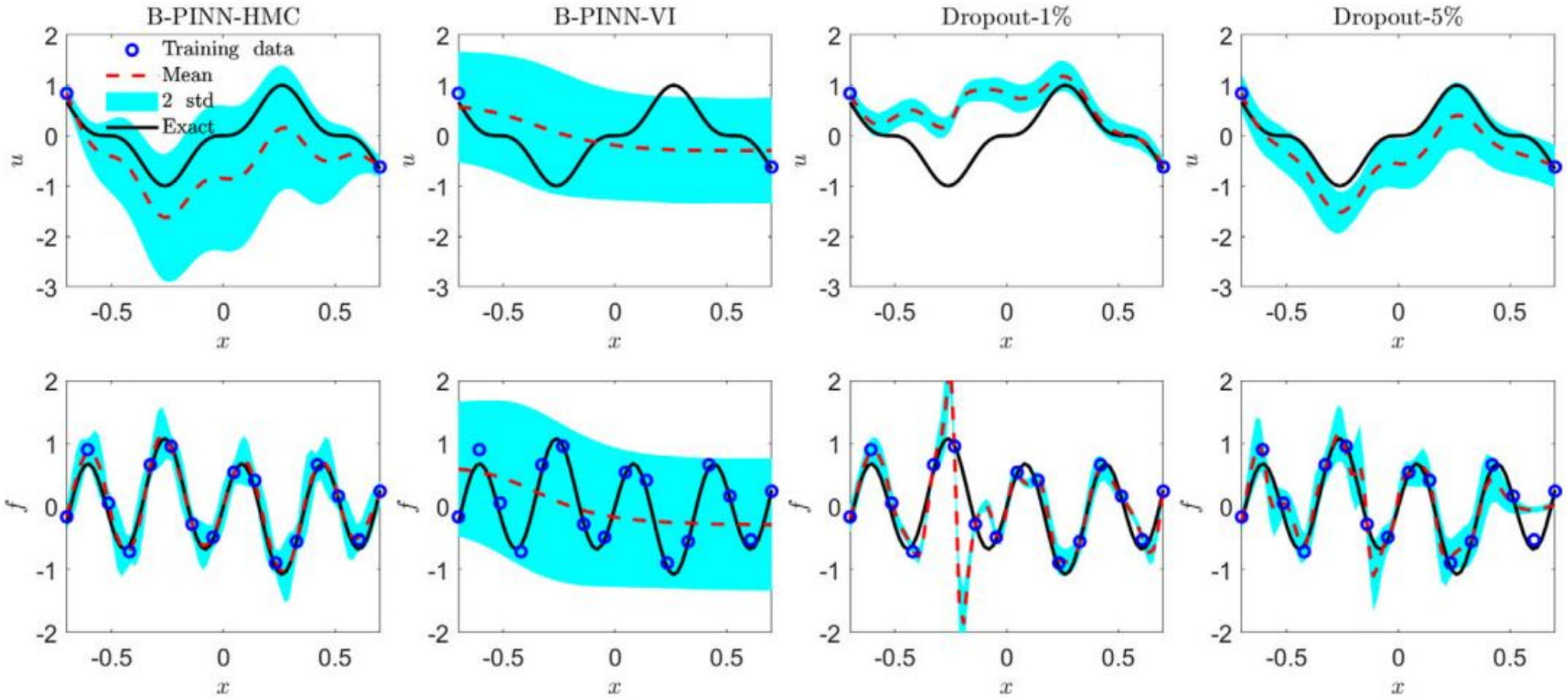


Fig. 3. Function approximatio … drop rate 1%, (e) Dropout with drop rate 5%, (f

Fig. 4. Function approximations for cases with larger noise. (a) $\epsilon_u \sim \mathcal{N}(0, 0.3^2)$. (b) $\epsilon_u \sim \mathcal{N}(0, 0.5^2)$.

$$\lambda \partial_x^2 u = f, x \in [-0.7, 0.7], \ \lambda = 0.01. \qquad u = \sin^3(6x)$$



(b)

Fig. 5. 1D linear Poisson equation - forward problem: predicted u and f from different methods with two data noise scales. (a) $\epsilon_f \sim \mathcal{N}(0, 0.01^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.01^2)$. (b) $\epsilon_f \sim \mathcal{N}(0, 0.1^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.1^2)$.

$$-\frac{\nu_e}{\phi}\partial_x^2 u + \frac{\nu u}{K} = g,\ x \in [0,1], \quad u = \frac{gK}{\nu}\left[1 - \frac{\cosh(r(x - H/2))}{\cosh(rH/2)}\right], \quad r = \sqrt{\frac{\nu\phi}{\nu_e K}}.$$

$$\nu_e = \nu = 10^{-3},\ \phi = 0.4,\ K = 10^{-3}.$$

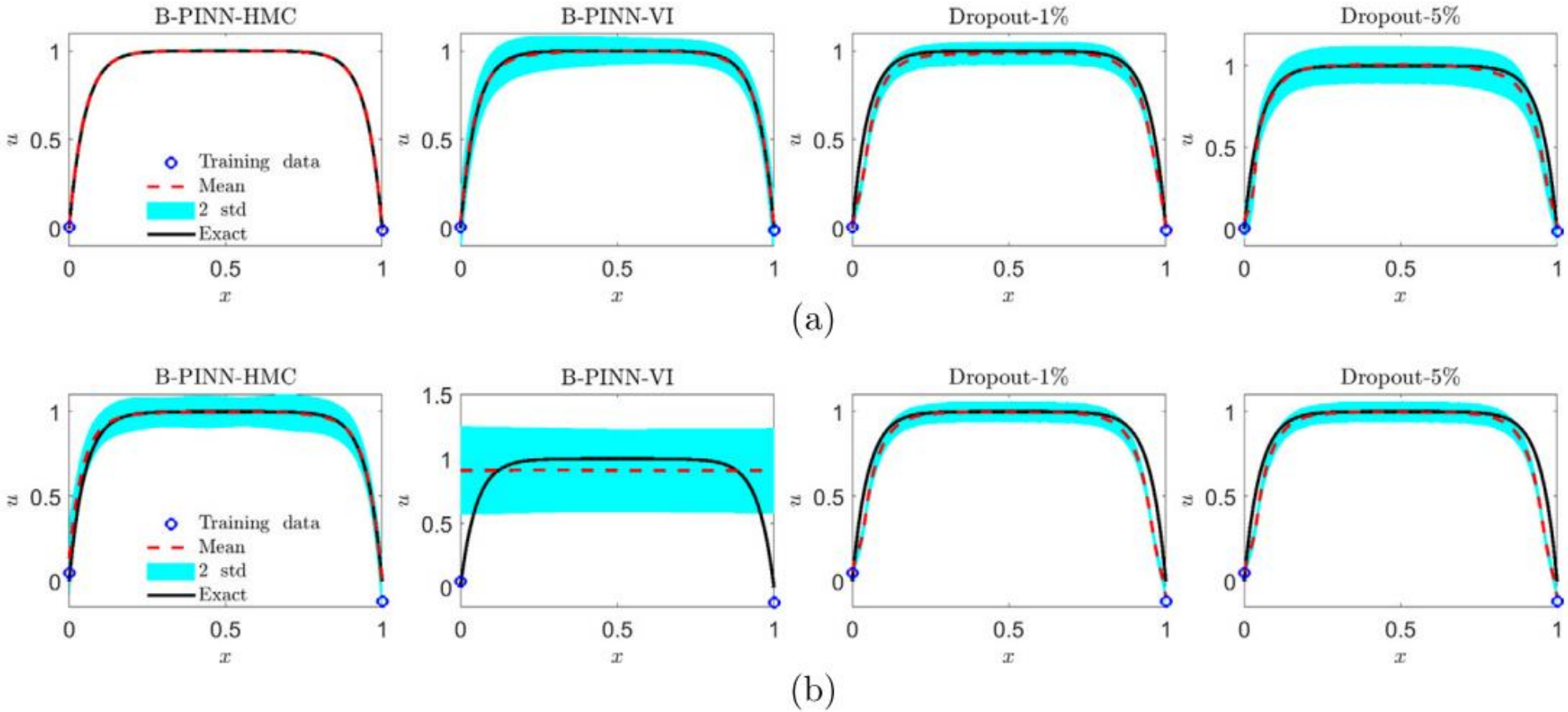


Fig. 7. 1D flow through porous media with boundary layer: predicted u from different methods with two data noise scales. (a) $\epsilon_g \sim \mathcal{N}(0, 0.01^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.01^2)$. (b) $\epsilon_g \sim \mathcal{N}(0, 0.1^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.1^2)$.

$$\lambda \partial_x^2 u + k \tanh(u) = f, \; x \in [-0.7, 0.7], \quad u = \sin^3(6x). \qquad \lambda = 0.01, \text{ and } k = 0.7$$

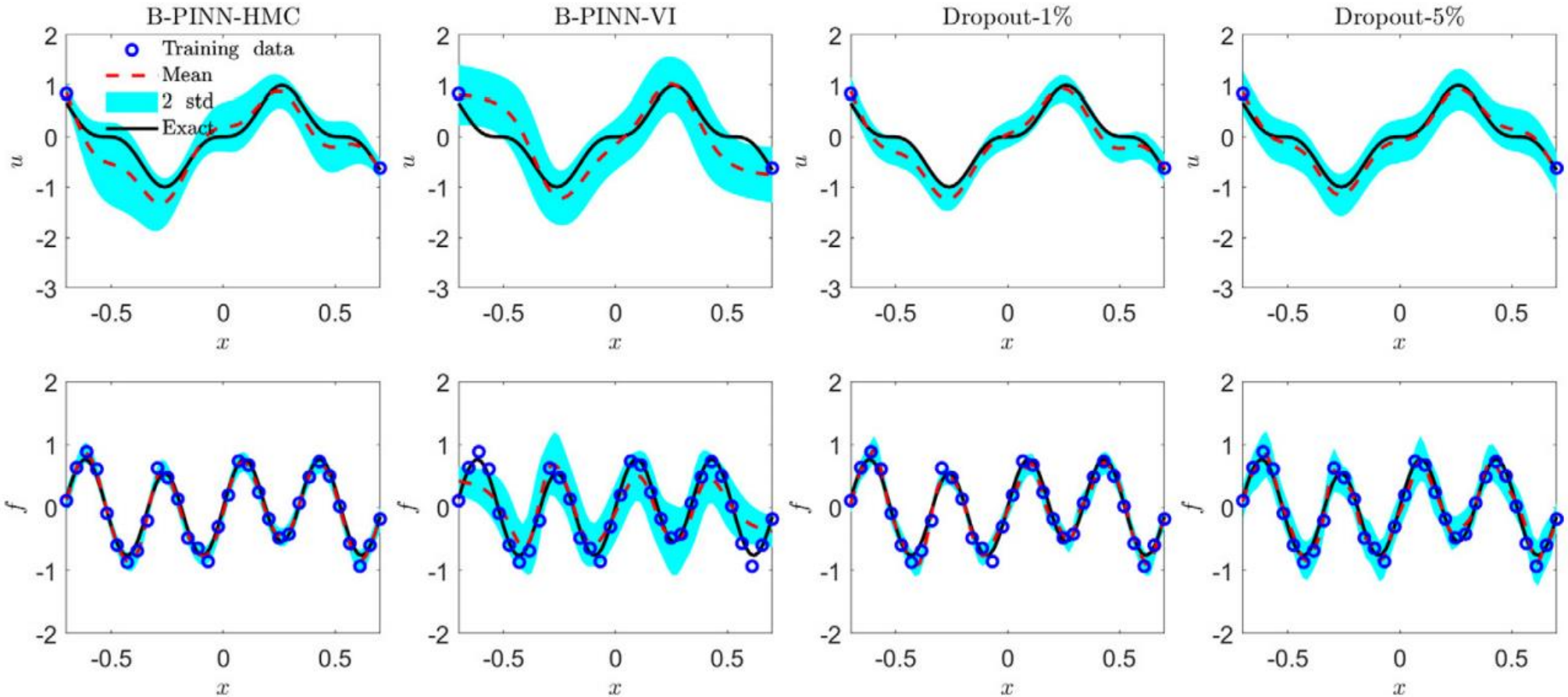
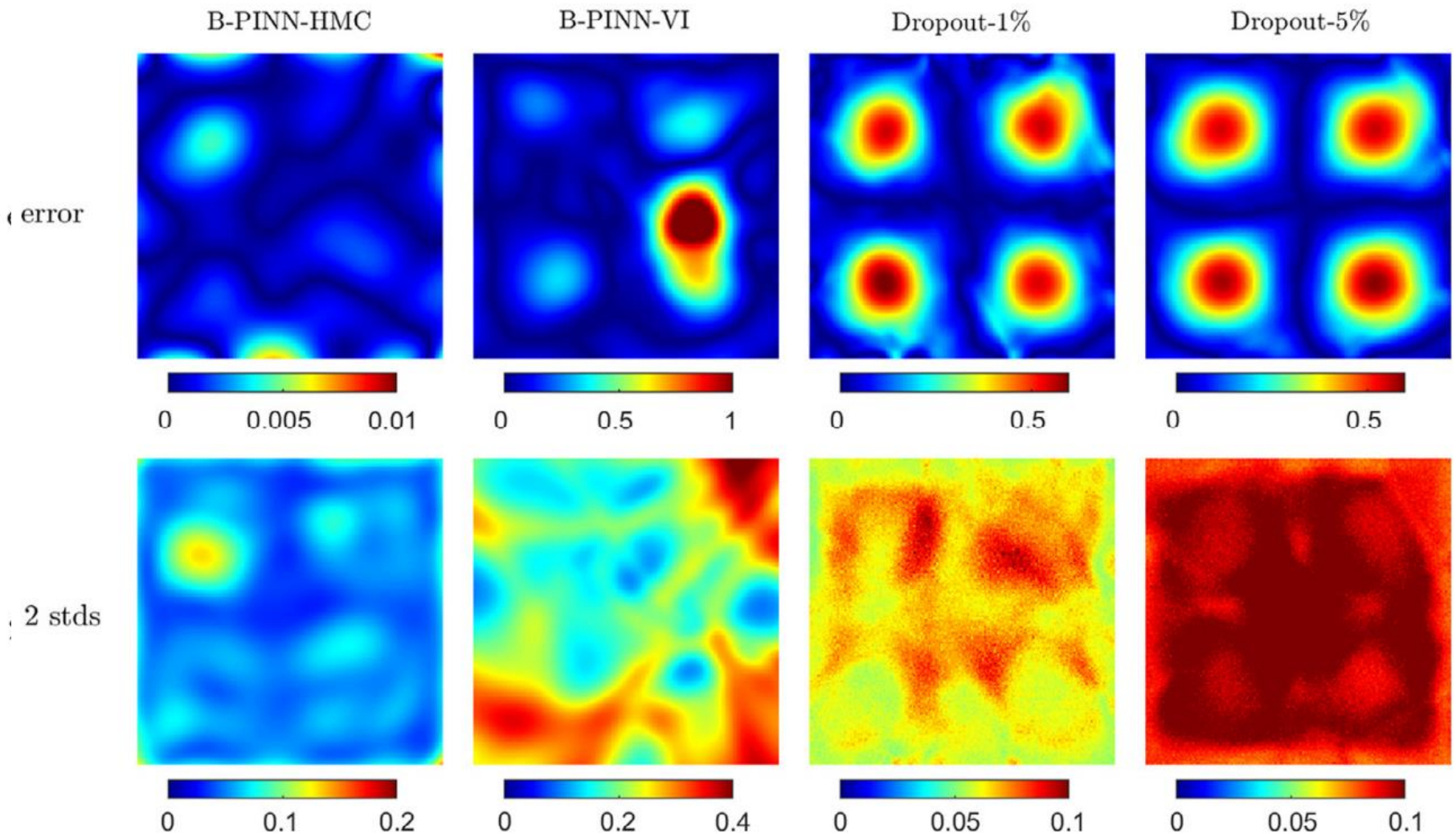


Fig. 8. 1D nonlinear Poisson equation - forward problem: predicted u and f from different methods with two data noise scales. (a) $\epsilon_f \sim \mathcal{N}(0, 0.01^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.01^2)$. (b) $\epsilon_f \sim \mathcal{N}(0, 0.1^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.1^2)$.

$$\lambda(\partial_x^2 u + \partial_y^2 u) + u(u^2 - 1) = f,\ x, y \in [-1, 1], \quad u = \sin(\pi x)\sin(\pi y).$$

$$\lambda = 0.01$$



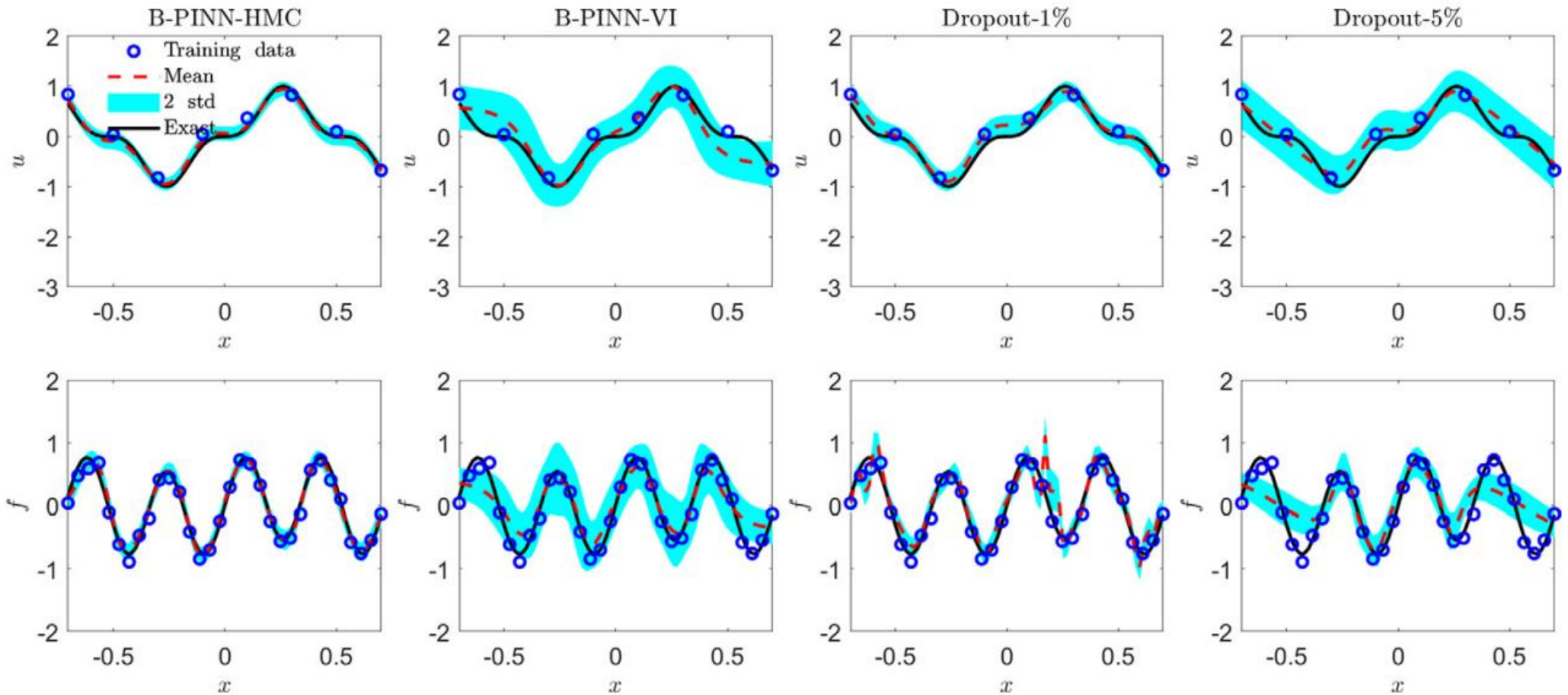
(b)

(a)

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

Table 1
1D diffusion-reaction system with nonlinear source term: Predicted mean and standard deviation for k using different uncertainty quantification methods. The 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}



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

Table 2
1D diffusion-reaction system with nonlinear source term: Predicted mean and standard deviation for k using different priors with exact value $k = 50$.

Prior for k		$N_{burnin} = 50,000$	$N_{burnin} = 100,000$
$k \sim \mathcal{N}(0, 1^2)$	Mean	42.77	43.27
	Std	1.39×10^{-1}	6.65×10^{-2}
$\log(k) \sim \mathcal{N}(0, 1^2)$	Mean	50.39	50.39
	Std	3.90×10^{-1}	3.96×10^{-1}

Table 3
1D diffusion-reaction system with nonlinear source term: Predicted mean and standard deviation for exact $k = 0.7$ and $k = 100$, using the prior $\log(k) \sim \mathcal{N}(0, 1^2)$.

Exact k		$N_{burnin} = 50,000$	$N_{burnin} = 100,000$
100	Mean	100.78	100.75
	Std	7.95×10^{-1}	8.08×10^{-1}
0.7	Mean	0.705	0.706
	Std	5.6×10^{-3}	5.2×10^{-3}

Comparison with PINNs

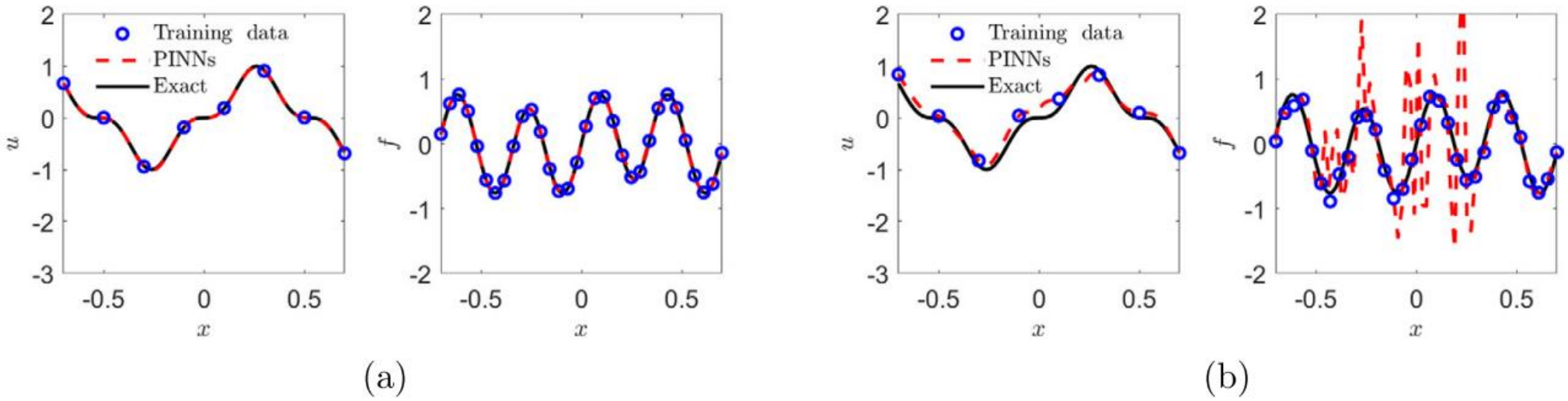


Fig. 12. 1D diffusion-reaction system with nonlinear source term (PINNs): Predicted u and f with two data noise scales. (a) $\epsilon_f \sim \mathcal{N}(0, 0.01^2)$, $\epsilon_u \sim \mathcal{N}(0, 0.01^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.01^2)$. (b) $\epsilon_f \sim \mathcal{N}(0, 0.1^2)$, $\epsilon_u \sim \mathcal{N}(0, 0.1^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.1^2)$.